



Des piles de sable aux automates de sable

Benoît Masson

► To cite this version:

Benoît Masson. Des piles de sable aux automates de sable. Modélisation et simulation. Université Nice Sophia Antipolis, 2006. Français. NNT: . tel-00144448

HAL Id: tel-00144448

<https://theses.hal.science/tel-00144448>

Submitted on 3 May 2007

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Des piles de sable aux automates de sable

THÈSE

présentée et soutenue publiquement le 13 décembre 2006

pour l'obtention du

Doctorat de l'Université de Nice – Sophia Antipolis
(spécialité Informatique)

par

Benoît MASSON

Composition du jury

Valérie BERTHÉ	(Directeur de recherches, LIRMM – Rapporteur)
Robert CORI	(Professeur, LIX – Rapporteur)
Bruno DURAND	(Professeur, LIF – Examineur)
Enrico FORMENTI	(Professeur, I3S – Directeur de thèse)
Jacques MAZOYER	(Professeur, LIP – Examineur)
Dominique ROSSIN	(Chargé de recherches, LIAFA – Rapporteur)
Michel RUEHER	(Professeur, I3S – Examineur)

Remerciements

Je tiens à remercier tout particulièrement mon directeur de thèse, Enrico FORMENTI, pour avoir toujours su m'encourager et me soutenir. Ses qualités humaines m'ont rendu ces trois années faciles à vivre.

Je voudrais également remercier Valérie BERTHÉ, Robert CORI et Dominique ROSSIN pour avoir pris le temps de relire mon manuscrit de thèse, et pour leurs remarques pertinentes. Je n'oublie pas les autres membres de mon jury, Bruno DURAND, Jacques MAZOYER et Michel RUEHER. Je les remercie de s'être libérés pour être présents le jour de la soutenance.

Enfin, j'ai une pensée pour tous ceux et celles qui m'ont accompagné pendant cette thèse, famille, amis, adversaires (sur le terrain de foot...) et collègues. Il serait trop long d'écrire vos noms ici, mais je ne vous oublie pas !

Sommaire

Introduction	1
Partie I Piles de sable	5
Chapitre 1 Bref historique	7
1.1 Définitions	8
1.1.1 Piles de sable	8
1.1.2 Systèmes de piles de sable	8
1.1.3 Graphe des orbites	9
1.2 Résultats connus	10
1.2.1 Point fixe	10
1.2.2 Longueur du transitoire	12
Chapitre 2 Piles de sable généralisées	15
2.1 Algorithme de calcul du point fixe	16
2.1.1 Découpage	16
2.1.2 Calcul	18
2.1.3 Fusion	21
2.2 Étude de l'algorithme	21
2.2.1 Exactitude	22
2.2.2 Complexité	23
2.3 Longueur du transitoire	24
Chapitre 3 Piles de sable symétriques	27
3.1 Définition de SSPM	28
3.2 Graphe des orbites	30
3.3 Points fixes	35
3.4 Longueur du transitoire	38
3.4.1 Longueur minimale du transitoire	38

3.4.2	À propos de la longueur maximale du transitoire	41
3.5	Perspectives	42
Partie II	Automates de sable	45
Chapitre 1	Définitions	47
1.1	Une topologie pour les configurations	48
1.2	Automates de sable	52
1.3	Relation avec les automates cellulaires	53
1.4	Caractérisation « à la Hedlund »	56
Chapitre 2	Dynamique des automates de sable	59
2.1	Propriétés ensemblistes	60
2.2	Automates conservateurs de grains	68
2.2.1	Définitions	69
2.2.2	Décidabilité	71
2.3	Automates ultimement périodiques et ultimement stables	74
2.3.1	Construction de l'automate de sable	74
2.3.2	Simulation de la machine à deux compteurs	76
2.3.3	Arrêt sur erreurs	79
2.3.4	Décidabilité	81
Conclusions et perspectives		83
Bibliographie		87
Annexes		91
Annexe A	Points fixes de (n) pour SSPM ($1 \leq n \leq 24$)	93
Index		95

Introduction

Depuis une dizaine d'années, l'utilisation des systèmes dynamiques discrets pour modéliser les phénomènes naturels complexes est de plus en plus fréquente. Plutôt que d'approximer les solutions à un problème continu (équations différentielles), on préfère désormais discrétiser ce problème. La puissance des ordinateurs actuels permet d'effectuer des simulations proches de la réalité, avec très peu de perte d'information. Parallèlement à cela, ces modèles sont définis de manière simple et sont bien formalisés mathématiquement, ce qui facilite leur étude. Les résultats présentés dans cette thèse s'inscrivent dans le cadre de l'étude formelle de tels systèmes. Plus précisément, ils concernent des systèmes dynamiques discrets obéissant à la notion de *Self-Organized Criticality* (SOC [3]). Il s'agit de systèmes dynamiques qui évoluent jusqu'à atteindre un état stable (ou point fixe), appelé « état critique ». Si une perturbation quelconque, même faible, vient modifier cet état critique, il se produit alors une réorganisation du système arbitrairement importante. Ensuite, le système évolue vers un nouvel état critique.

Ce phénomène se retrouve dans un grand nombre de systèmes naturels (voir [49] par exemple). Le *Chip Firing Game* (CFG [46]) est l'une des ses toutes premières formalisations en tant que système dynamique discret. Un CFG est un graphe orienté dont chaque sommet v est associé à un nombre de jetons c_v et à un seuil δ_v . Si $c_v \geq \delta_v$, alors v « perd » δ_v jetons qui sont répartis équitablement entre les s_v successeurs de v (on pose en général $\delta_v = s_v$ afin de ne pas avoir à fractionner les jetons). Si un tel système a convergé vers un état stable et que l'on ajoute des jetons à un des sommets, ceux-ci vont alors se répartir dans le graphe en un nombre indéterminé d'étapes.

Le modèle de pile de sable qui sert de point de départ à notre étude est un exemple particulier de CFG et s'appelle *Sand Pile Model* (SPM [3, 27]). Pour SPM, le graphe est linéaire, tous les sommets ont deux voisins qui sont à la fois successeurs et prédécesseurs. Pour chaque sommet v , $\delta_v = 2$. Chaque sommet du graphe peut être vu comme une « colonne » de grains, le nombre de jetons du sommet correspondant à la différence entre le nombre de grains de la colonne et celui de sa voisine de droite (voir figure 1). En pratique, lorsque la différence de grains vaut 2 ou plus, la colonne de gauche perd un grain qui va sur sa voisine de droite : on dit qu'il y a *avalanche*. Ce modèle, créé à l'origine pour modéliser la formation des piles de sables ([3]), se retrouve également dans la théorie des partitions d'entiers ([7]). Il illustre parfaitement le phénomène SOC, car si l'on ajoute un grain à une pile de sable stable, celui-ci va provoquer un nombre variable d'avalanches avant que la pile ne se stabilise à nouveau.

Le modèle SPM et ses variantes ont donné lieu à de nombreux travaux [7, 27, 44, 17, 42, 41, 43, 13, 39, 40, 14, 15, 30, 29, 28]. Parmi ces variantes, citons l'ensemble de modèles $\text{IPM}(k)$ (*Ice Pile Model* [29]). Dans ce cas, on ajoute la possibilité que les grains « glissent » sur des plateaux de taille strictement inférieure à k . Remarquons que SPM est égal à $\text{IPM}(1)$. Une autre variante intéressante de ces modèles, moins étudiée cependant (voir quand même [27, 22]), concerne le cas où tous les grains se déplacent en même temps, on appelle cela le mode parallèle. En termes

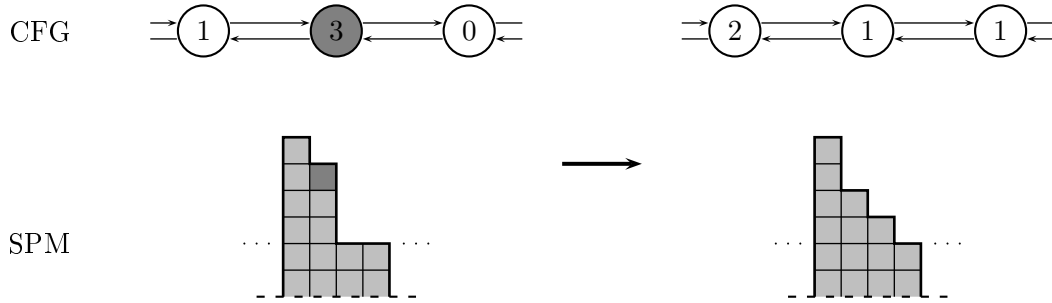


FIG. 1 – Relation entre CFG et SPM. Les poids des sommets du CFG correspondent aux différences de hauteur entre deux colonnes successives de SPM. La case grisée est celle qui est modifiée.

de CFG, cela veut dire que tous les sommets v tels que $c_v \geq \delta_v$ perdent leurs δ_v grains pendant le même pas de temps. Cela rend le modèle déterministe, car il n'y a plus à choisir quel sommet faire évoluer en premier, et plus réaliste. En effet, dans la nature, les grains d'une pile de sable tombent dès que la pente de la pile le leur permet, sans attendre leur tour.

La dynamique des modèles de piles de sable SPM et $\text{IPM}(k)$ est bien connue [27, 22, 43, 39, 40, 30, 29, 28]. On sait que le système a une dynamique de type point fixe, c'est-à-dire qu'il atteint toujours un état stable. On peut définir le *graphe des orbites* comme le graphe orienté dont les sommets sont les piles de sables accessibles, et où une pile est le successeur d'une autre si elle peut être obtenue de la première à l'aide d'une avalanche. À partir d'une colonne unique de grains, ce graphe forme un treillis. Les treillis représentent des ordres partiels [4], dans notre cas ils permettent de montrer que l'état stable d'une pile de sable est unique, quel que soit l'ordre dans lequel sont effectuées les avalanches. On connaît également le nombre d'itérations nécessaires pour obtenir cet état stable à partir d'une configuration initiale où tous les grains sont groupés dans la même colonne.

L'objectif principal de cette thèse était d'étendre ces résultats, selon deux directions principales. Dans un premier temps, nous avons essayé de généraliser l'étude des piles de sable en partant des modèles existants. Dans un second temps, nous introduisons un nouveau modèle, les *automates de sable* dont l'intérêt est double : ils unifient tous les modèles locaux de piles de sable en un seul, et à l'aide d'un cadre formel solide ils simplifient l'étude générale de leur dynamique.

Dans une première partie intitulée « piles de sable », nous récapitulons tous nos résultats concernant la généralisation des piles de sable. Après avoir rappelé brièvement les définitions, notations et résultats que nous utilisons, nous étudions un algorithme permettant de calculer rapidement le point fixe d'une pile de sable quelconque, où les grains ne sont pas nécessairement concentrés dans une seule colonne. Il est impossible de trouver une formule close pour décrire le point fixe dans le cas général, notre algorithme effectue le calcul en temps $\mathcal{O}(l \cdot \min(n, l))$ (n est le nombre de grains de la pile initiale, l est sa longueur). Le principe de base de cet algorithme est le suivant : la pile initiale est divisée en sous-piles pour lesquelles les formules connues s'appliquent ; puis, à l'aide de fusions successives, le point fixe final est construit pas à pas. Nous montrons également que même en partant d'une configuration initiale quelconque, le graphe des orbites est encore un treillis pour tous les modèles $\text{IPM}(k)$ – donc, en particulier, pour SPM.

Cette partie se termine sur l'introduction d'un nouveau modèle de piles de sable, *Symmetric*

Sand Pile Model (SSPM). Ce modèle vise à rendre la simulation des piles de sable plus réaliste en permettant aux grains de tomber selon plusieurs directions, et non une seule comme les précédents. Nous définissons SSPM par les règles suivantes : un grain peut tomber vers la gauche si la différence de hauteur entre sa colonne et sa voisine de gauche excède 2, de même si la différence vers la droite est supérieure ou égale à 2 un grain peut se déplacer vers la droite. Ce modèle n'est évidemment pas déterministe, car un même grain peut parfois se déplacer à la fois vers la gauche et vers la droite. Le graphe des orbites n'est donc plus un treillis ; cependant il conserve une structure simple. Nous montrons que dans le cas d'une pile initiale de n grains répartis sur une seule colonne, tous les sommets du graphe des orbites peuvent être caractérisés formellement. De plus on montre qu'il y a toujours exactement $\lfloor \sqrt{n} \rfloor$ points fixes. Ce résultat surprenant ouvre la voie à une nouvelle étude des piles de sable multi-directionnelles et multi-dimensionnelles. Nous concluons en évaluant le temps nécessaire pour obtenir l'un de ces points fixes, ce temps est du même ordre de grandeur que celui obtenu pour SPM. L'ajout d'une direction nous permet donc un gain important en termes de « réalisme », sans perte d'efficacité computationnelle.

Dans la seconde partie, nous introduisons les « automates de sable » [8]. Ce système dynamique discret est défini de manière semblable aux automates cellulaires [37]. Il agit sur une grille discrète \mathbb{Z}^d , à l'aide d'une table de transition finie (la *règle locale*). Chaque point de la configuration évolue en fonction de son voisinage. Pour simuler les différents modèles de piles de sable, il suffit ainsi d'entrer la table de transition correspondante. Remarquons que les automates de sable ont un fonctionnement synchrone (tous les points sont transformés en même temps), c'est donc le mode parallèle de ces modèles qui est simulé.

Nous commençons par introduire une *topologie* associée à l'espace des configurations. Avec cette topologie, l'espace a des propriétés fortes qui nous aident dans notre étude. Par exemple, sans être compact, il est localement compact et donc complet. Nous rappelons que les automates de sable sont exactement les fonctions continues de cet espace dans lui-même qui présentent les propriétés suivantes : invariance horizontale et verticale (*i.e.* l'automate commute avec les décalages horizontaux et verticaux), conservation des infinis (une valeur infinie n'est pas modifiée). Ce modèle bénéficie d'un formalisme mathématique qui permet d'obtenir des résultats plus généraux que ceux obtenus avec des méthodes algébriques et combinatoires pour les piles de sable. Enfin, nous verrons que les automates de sable sont aussi puissants que les machines de Turing, puisqu'ils sont capables de simuler les automates cellulaires.

Dans un second temps, on s'intéresse à la dynamique des automate de sable. Nous étudions des propriétés classiques de la théorie du chaos déterministe, telles que la surjectivité et l'injectivité des automates, en montrant les relations existant entre elles. Puis, nous continuons avec des propriétés dynamiques fortes : la conservation des grains et l'ultime périodicité. Nous cherchons à identifier les automates de sable dont la dynamique s'approche de celles des piles de sable réelles, la notion de conservation de grains est donc très importante : un automate conserve les grains si entre deux pas de temps le nombre total de grains de la configuration n'est pas modifié, *i.e.* il n'y a eu que des déplacements, pas de destruction ni de création de grains. De même, le fait de savoir à l'avance si le comportement d'un automate va se stabiliser (complètement, ou simplement de manière périodique) est important pour isoler les automates donnant lieu à des avalanches sans fin. Ces automates sont dits ultimement stables lorsque les configurations n'évoluent plus au bout d'un certain temps, ou ultimement périodiques lorsque leur comportement devient périodique. Nous montrons qu'il est possible de décider si un automate de sable donné conserve les grains, mais également qu'il est impossible de décider si un automate de sable est ultimement stable ou ultimement périodique. Les résultats de ce chapitre ne font que confirmer la complexité de la

dynamique des automates de sable, aussi variée qu'imprévisible.

Pour terminer, nous concluons avec de nouvelles perspectives de recherche pour étendre les résultats présentés dans cette thèse. D'une part, nous proposons plusieurs pistes pour compléter nos résultats : extensions du modèle SSPM, autres propriétés dynamiques des automates de sable, *etc.* D'autre part, nous introduisons des perspectives nouvelles et originales concernant différents aspects de ces travaux. Nous pourrions par exemple rechercher de nouveaux modèles de piles de sable en modifiant les hypothèses de départ, par exemple en changeant la structure du support, pour obtenir des résultats encore plus réalistes. Mais surtout, les automates de sable sont un système complexe entièrement nouveau, et pour simplifier et clarifier leur étude une classification des automates selon leur comportement serait utile. Il serait alors intéressant de voir dans quelle classe se trouveraient les automates conservateurs de grains ou ultimement périodiques, et les modèles de piles de sable en général. Un autre aspect de ces systèmes qui mériterait d'être traité est leur puissance en termes de calcul. Que peuvent calculer les automates de sable, et à quelle vitesse ? Cette thèse s'efforce de démontrer que ce système dynamique discret est original pour sa dynamique, il l'est certainement dans ses capacités computationnelles.

Première partie

Piles de sable

1

Bref historique

Sommaire

1.1 Définitions	8
1.1.1 Piles de sable	8
1.1.2 Systèmes de piles de sable	8
1.1.3 Graphe des orbites	9
1.2 Résultats connus	10
1.2.1 Point fixe	10
1.2.2 Longueur du transitoire	12

Dans cette partie de la thèse nous présentons les travaux effectués dans le cadre des *piles de sable*. Nous appelons piles de sable les états obtenus à partir des modèles SPM (*Sand Pile Model* [3, 27, 29, 30]) et IPM(k) (*Ice Pile Model* [7, 29]). Ces modèles sont utilisés notamment pour simuler la formation des tas de sable, après une avalanche de grains ou n'importe quelle autre perturbation du tas, selon le principe des systèmes SOC (*Self-Organized Criticality* [3]) : à partir d'une configuration initiale quelconque, le système évolue jusqu'à atteindre un « état critique ». Toute perturbation de cet état critique entraîne une réorganisation complète du système, jusqu'à obtenir un nouvel état critique. Les piles de sable illustrent parfaitement ce phénomène, en effet, chaque tas de sable « stable » est modifié si de nouveaux grains arrivent ou si d'autres disparaissent.

Les différents modèles étudiés consistent en un tas de sable initial discrétisé, dans lequel chaque grain est représenté par un carré dans une grille à 2 dimensions, et en deux règles locales très simples. Une règle verticale déplace les grains verticalement (ils tombent) lorsqu'il y a un trou, et une règle horizontale leur permet de glisser sur des plateaux. Tout ceci est formalisé dans la partie 1.1 pour les modèles SPM et IPM(k). Ces modèles ont été particulièrement étudiés [7, 27, 44, 17, 42, 41, 43, 15], et leur comportement bien identifié [39, 40, 30, 29, 28]. On sait notamment qu'il existe un unique état critique (ou état stable, ou encore de manière impropre, point fixe) que l'on peut décrire ; on connaît également les états transitoires et le nombre d'itérations nécessaires à l'obtention du point fixe, *etc.* (voir partie 1.2).

Le problème principal de ces modèles est leur manque de généralité. D'une part, les seules configurations initiales considérées sont les configurations dont tous les grains sont concentrés dans une seule colonne, d'autre part ces modèles sont uni-directionnels, uni-dimensionnels, et le passage aux dimensions supérieures n'est pas immédiat. Le premier problème est traité dans le chapitre 2, où l'on considère des configurations initiales quelconques. Pour ce qui est de la

généralisation des modèles eux-mêmes et du passage aux dimensions supérieures, une étude de modèles symétriques est faite dans le chapitre 3. Le chapitre courant se contente de définir les notions nécessaires au travail sur les piles de sable, puis effectue un petit récapitulatif des résultats connus.

1.1 Définitions

1.1.1 Piles de sable

Une *pile de sable* (ou *configuration*) de *longueur* l est une suite finie d'entiers positifs $c = (c_1, c_2, \dots, c_l)$. Chaque c_i est appelé *colonne* de c . Le nombre $n = \sum_{i=1}^l c_i$ est le *nombre de grains* de la pile. Soit $\mathfrak{C} = \cup_{k \in \mathbb{N}^*} (\mathbb{N})^k$ l'ensemble des configurations sur \mathbb{N} .

Étant donnée une pile de sable $c = (c_1, \dots, c_l)$, on dit que c contient un *plateau* (c_i, \dots, c_j) à la position i si $c_h = c_{h+1}$ pour tout $i \leq h < j$. Le nombre $k = j - i + 1$ est la *longueur* du plateau, et $p = c_i$ est sa *hauteur*, il est alors noté $p^{[k]}$. La configuration c contient une *falaise* (c_i, c_{i+1}) à la position i si $c_i \geq c_{i+1} + 2$.

On représente de manière naturelle une pile de sable (c_1, \dots, c_l) sur une grille bi-dimensionnelle, où c_i est le nombre de grains de la i^{e} colonne représenté par c_i cases verticales sur la grille (figure 2). Ce dessin est appelé *diagramme de Ferrers* dans la théorie des partitions [2].

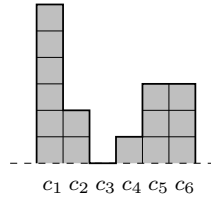


FIG. 2 – Représentation graphique de la configuration $(6, 2, 0, 1, 3, 3)$. Elle contient trois falaises (c_1, c_2) , (c_2, c_3) et (c_6, c_7) , et un plateau en (c_5, c_6) .

1.1.2 Systèmes de piles de sable

Un *système* de piles de sable (parfois appelé *modèle*) est un ensemble fini de règles qui vont agir sur les configurations et les transformer. Le plus connu est *SPM* (*Sand Pile Model*), introduit dans [3, 27]. SPM consiste en une règle locale unique, la *règle verticale* : s'il y a une falaise, alors un grain tombe de la colonne de gauche sur sa voisine de droite.

Formellement, si pour une configuration $c = (c_1, \dots, c_l)$ il existe un i tel que $c_i - c_{i+1} \geq 2$ (voir figure 3), c peut évoluer en c' défini par :

$$\begin{cases} c'_i &= c_i - 1 \\ c'_{i+1} &= c_{i+1} + 1 \\ c'_j &= c_j \end{cases} \quad \text{pour } j \notin \{i, i+1\} .$$

Dans cette partie, SPM est appliqué à des configurations initiales qui contiennent n grains dans la première colonne et rien ailleurs, en d'autres termes elles sont de type (n) .

SPM peut être généralisé par l'ensemble de modèles $L(\theta)$, $\theta \in \mathbb{N}^*$, définis dans [43]. Les modèles $L(\theta)$ possèdent la même règle verticale, mais celle-ci ne peut être appliquée que lorsque

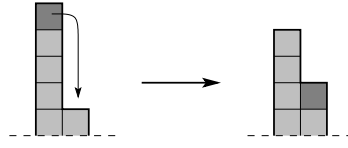


FIG. 3 – Règle verticale de SPM pour une falaise de hauteur 4.

$c_i - c_{i+1} > \theta$ (donc SPM est $L(1)$). Comme nous nous intéressons exclusivement à la dynamique des systèmes, ces modèles n'apportent rien de plus que SPM et nous ne les traitons pas.

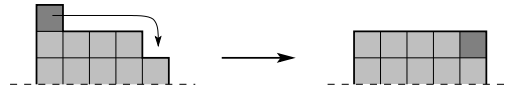
L'autre système qui nous intéresse est l'ensemble des modèles $IPM(k)$ (*Ice Pile Model*, introduits dans [7, 29]), pour $k \in \mathbb{N}^*$. Ils possèdent la règle verticale de SPM ainsi qu'une *règle horizontale* qui permet aux grains de « glisser » sur un plateau de longueur inférieure à k . Intuitivement, alors que SPM et $L(\theta)$ modélisent les tas de sable se stabilisant sur une pente de 45° ou plus, les modèles $IPM(k)$ permettent d'obtenir des tas de pentes inférieures. Là aussi, la configuration initiale est réduite à une unique colonne de n grains.

Si pour une configuration $c = (c_1, \dots, c_l)$, il existe $i \leq j$ tels que

$$c_i = c_{i+1} + 1 = c_{i+2} + 1 = \dots = c_j + 1 = c_{j+1} + 2 ,$$

avec $j - i < k$ (illustré figure 4), alors c peut évoluer en c' défini par :

$$\begin{cases} c'_i &= c_i - 1 \\ c'_{j+1} &= c_{j+1} + 1 \\ c'_h &= c_h \end{cases} \quad \text{pour } h \notin \{i, j+1\} .$$

FIG. 4 – Règle horizontale de $IPM(k)$ sur un plateau de longueur 3.

Plus k est important, plus les grains peuvent glisser loin. En particulier, SPM est exactement le modèle $IPM(1)$. Selon les cas, SPM est traité comme un cas particulier de $IPM(k)$ ou comme une première étape. En effet, les résultats sont souvent plus simples à exprimer et à prouver pour SPM, et sont donc un bon point de départ à l'étude plus générale de $IPM(k)$.

On peut également définir le modèle $IPM(\infty)$, que nous n'utilisons pas car il n'est pas défini de manière locale (il faut pouvoir connaître un nombre non borné de valeurs pour décider quoi faire). C'est lui qui a été créé par Brylawski [7] avec la notation L_B , à l'origine pour étudier les *partitions d'entiers* [2, 31] : étant donné un entier $n \in \mathbb{N}$, l'ensemble des partitions de n en entiers plus petits correspond exactement à l'ensemble des configurations obtenues par $IPM(\infty) = L_B$.

1.1.3 Graphe des orbites

Une fois le modèle défini, il y a plusieurs façons de faire fonctionner le système. En mode *séquentiel* (ou *asynchrone*), un seul grain se déplace à la fois, et quand il y a plusieurs possibilités on choisit n'importe laquelle. Le modèle n'est donc pas déterministe, mais on verra que ce n'est pas très important. En mode *parallèle* (ou *synchrone*), tous les grains pouvant tomber le font

en même temps, ce qui lève les ambiguïtés. Curieusement, bien que le mode parallèle soit plus réaliste, il a été moins étudié que le séquentiel.

Le système évolue donc selon l'un des deux modes, à partir de la configuration initiale jusqu'à obtenir un *point fixe* (ou *état stable*), c'est-à-dire une configuration n'ayant plus aucune falaise.

Une suite de configurations $(c^i)_{i \in [0, t]}$ est appelée une *orbite* de la configuration initiale c^0 si pour tout $i \in [0, t - 1]$, c^{i+1} peut être obtenue à partir de c^i en appliquant une des règles du système.

L'ensemble des orbites de même configuration initiale c peut se représenter par le *graphe des orbites* de c , noté $\mathcal{G}_c = \langle V, E \rangle$. L'ensemble des sommets V est l'ensemble des configurations appartenant à une orbite de c , et $(c^1, c^2) \in E$ (avec $E \subseteq V \times V$) si et seulement si c^2 peut être obtenue en appliquant une des règles du système quelque part dans c^1 .

Les figures 5 illustrent les graphes des orbites de la configuration (7) pour le système IPM(3) (glissement sur des plateaux de longueur au plus 2). Les grains en gris foncé sont ceux qui sont susceptibles de se déplacer ; sur la figure 5(a) qui représente le mode séquentiel, un seul se déplace à la fois ; sur la figure 5(b) représentant le mode parallèle, tous tombent en même temps.

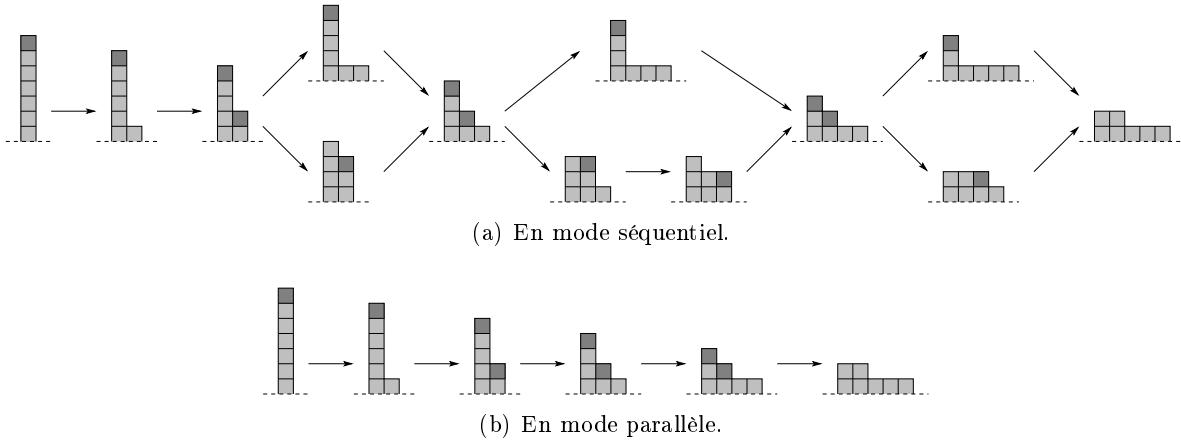


FIG. 5 – Graphes des orbites de (7) pour IPM(3).

Par abus de langage, on dit naturellement qu'une configuration c appartient à un graphe d'orbite $\langle V, E \rangle$ lorsque $c \in V$, c est alors *atteignable*. De plus, une configuration $c \in V$ est appelée *point fixe* du graphe si aucune arête de E ne part de c .

On peut d'ailleurs remarquer sur les exemples figure 5 que l'on obtient toujours le même point fixe, quel que soit le mode et la priorité considérée. Ce résultat est normal, on verra dans la partie suivante que le point fixe est toujours unique.

1.2 Résultats connus

1.2.1 Point fixe

On présente ici les résultats généraux pour IPM(k), tels qu'ils ont été démontrés dans [29].

Théorème 1 ([29]) *Pour tous les systèmes IPM(k), pour tout $n \in \mathbb{N}^*$, le graphe des orbites $\mathcal{G}_{(n)}$ est un treillis fini.*

Ce résultat majeur implique que toute configuration initiale de type (n) aboutit à un point fixe, et que celui-ci est toujours le même indépendamment de l'ordre dans lequel sont effectuées les transitions. Avec le mode parallèle, qui correspond au mode séquentiel où on choisit d'appliquer la règle le plus à droite possible, on obtient donc le même point fixe.

Le cas particulier de SPM avait déjà été démontré auparavant dans [27].

Lemme 2 ([29]) *Soit c une configuration contenant n grains, de longueur l . Pour tout $k \in \mathbb{N}^*$, $c \in \mathcal{G}_{(n)}$ pour $\text{IPM}(k)$ si et seulement si pour tous $1 \leq i < j \leq l$, on a $j - i \leq k(c_i - c_j + 1)$.*

Remarque 1 *Il faut noter que dans [29], ce résultat est présenté de manière équivalente mais plus complexe. La condition ci-dessus suffit en effet à garantir le fait que pour $\text{IPM}(k)$, entre deux plateaux de longueur $k+1$ (la plus grande possible), il y a soit une falaise soit un plateau de longueur strictement inférieure à k . Dans ce cas, un grain peut se déplacer et un des plateaux de taille $k+1$ peut être réduit.*

Ce point de vue correspond exactement à un autre énoncé du lemme pour le cas particulier SPM : une configuration c appartient à un graphe si et seulement si entre deux plateaux quelconques de c , il y a au moins une falaise.

La figure 6 présente deux exemples de configurations à 14 grains pour $\text{IPM}(2)$: la configuration de la figure 6(a) est valide et appartient à $\mathcal{G}_{(14)}$, la configuration de la figure 6(b) ne l'est pas : entre les colonnes 2 et 7, il y a 4 colonnes au lieu des $2 * (2 - 1 + 1) - 1 = 3$ autorisées.

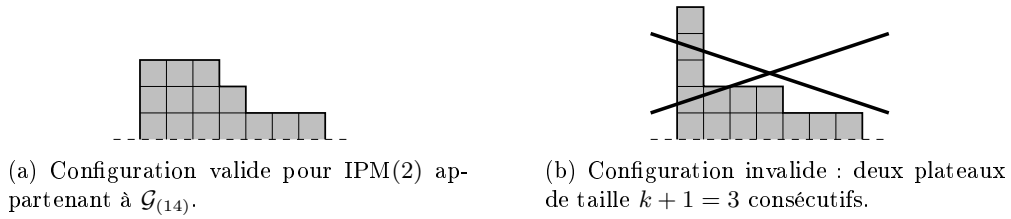


FIG. 6 – Exemples de configurations à 14 grains dans $\text{IPM}(2)$.

Ce lemme permet de caractériser simplement le point fixe π obtenu par un modèle $\text{IPM}(k)$. Étant donné qu'il ne peut y avoir aucun déplacement dans un point fixe, π ne contient que des plateaux de longueur k , sauf au sommet où il peut y avoir un plateau de longueur $k' < k$, et au plus un plateau de longueur $k+1$.

Pour formaliser ceci, on a besoin de nouvelles variables. Soit p l'unique entier vérifiant

$$\frac{1}{2}p(p+1)k \leq n < \frac{1}{2}(p+1)(p+2)k ,$$

et k' et $p' < p+1$ définis de manière unique par

$$n - \frac{1}{2}p(p+1)k = k'(p+1) + p' .$$

Le triplet $\langle p, p', k' \rangle$ est appelé *décomposition entière* de n . Alors le point fixe de $\text{IPM}(k)$ est caractérisé par la proposition suivante.

Proposition 3 ([29]) *Pour tous les systèmes $\text{IPM}(k)$, pour tout $n \in \mathbb{N}^*$, le point fixe π du graphe $\mathcal{G}_{(n)}$ est défini par :*

$$\pi = \begin{cases} (p+1)^{[k']}, p^{[k]}, (p-1)^{[k]}, \dots, 1^{[k]} & \text{si } p' = 0, \\ (p+1)^{[k']}, p^{[k]}, (p-1)^{[k]}, \dots, (p'-1)^{[k]}, p'^{[k+1]}, (p'-1)^{[k]}, \dots, 1^{[k]} & \text{sinon,} \end{cases}$$

où $\langle p, p', k' \rangle$ est l'unique décomposition entière de n telle que $n = \frac{1}{2}p(p+1)k + (p+1)k' + p'$.

À titre d'exemple, la figure 7 représente le point fixe de $\mathcal{G}_{(18)}$ pour IPM(2). Comme prévu par le lemme 2, il n'y a qu'un plateau de taille $k+1=3$.

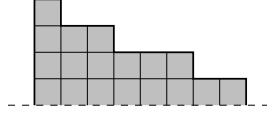


FIG. 7 – Point fixe de $\mathcal{G}_{(18)}$ pour IPM(2). La décomposition entière de 18 est $\langle 3, 2, 1 \rangle$, on a bien $18 = \frac{1}{2} \cdot 3 \cdot 4 \cdot 2 + 4 \cdot 1 + 2$.

1.2.2 Longueur du transitoire

La *longueur du transitoire* d'une configuration initiale c est le nombre de fois qu'une règle est appliquée à c avant d'obtenir un point fixe, autrement dit la longueur dans \mathcal{G}_c d'un chemin de c à son point fixe π . Cette longueur dépend tout d'abord du mode de fonctionnement utilisé, mais aussi dans le mode séquentiel il existe généralement plusieurs chemins pour parvenir au point fixe (voir figure 5(a)). Ces chemins n'ont alors pas forcément une longueur identique.

Mode séquentiel

Pour contourner ce problème de chemins multiples, deux objets ont été créés pour le mode *séquentiel*. Les *HV-chaînes* [31] sont les chemins consistant en l'application de la règle verticale uniquement, puis de la règle horizontale uniquement. Les *chaînes modulaires* [27] sont les chemins où la règle horizontale n'est autorisée qu'en cas d'impossibilité d'appliquer la règle verticale.

Les théorèmes suivants caractérisent les chemins les plus longs pour obtenir le point fixe.

Théorème 4 ([31]) Dans $L_B = IPM(\infty)$, les HV-chaînes entre deux configurations quelconques d'un graphe des orbites ont toute la même longueur, et celle-ci est maximale.

Théorème 5 ([29]) Pour tous les systèmes IPM(k), les chaînes modulaires entre deux configurations d'un graphe des orbites sont toutes de longueur maximale.

Cette dernière proposition permet de calculer la longueur maximale du transitoire en mode séquentiel $t_{seq}(k)$, pour tous les systèmes IPM(k) [29].

Proposition 6 ([29]) Pour tous les systèmes IPM(k), pour tout $n \in \mathbb{N}^*$, la longueur maximale du transitoire en mode séquentiel $t_{seq}(k)$ de la configuration (n) est

$$t_{seq}(k) = 2 \binom{q}{3} + qq' - k' \binom{p+1}{2} - k \binom{p+1}{3} - \binom{p'}{2} = \mathcal{O}(n^{3/2}) ,$$

où $\langle p, p', k' \rangle$ est l'unique décomposition entière de $n = \frac{1}{2}p(p+1)k + (p+1)k' + p'$, et $\langle q, q' \rangle$ avec $q' \leq q$ est l'unique décomposition de n en $n = \frac{1}{2}q(q+1) + q'$ (décomposition entière de n pour $k=1$).

En particulier pour SPM, comme $k=1$, $k'=0$, $q=p$ et $q'=p'$, on a $t_{seq}(1) = \frac{1}{6}p(p-1)(p+1) + \frac{1}{2}p'(2p+1-p')$.

Remarque 2 *Étant donné que pour SPM il n'y a pas de règle horizontale, dans ce modèle tous les chemins d'une configuration à une autre ont exactement la même longueur. La longueur du transitoire est donc toujours donnée de manière exacte par $t_{seq}(1)$.*

Dans le cas général par contre, la longueur minimale du transitoire est inconnue, même s'il est vraisemblable qu'elle est aussi en $\mathcal{O}(n^{3/2})$.

Mode parallèle

Curieusement, l'étude de la longueur du transitoire pour le mode parallèle $t_{par}(k)$ est plus difficile. Il n'existe un résultat que pour le modèle SPM. Dans [27] on trouve un premier encadrement pour $t_{par}(1)$, l'ordre de grandeur définitif n'étant prouvé que dans [22].

Proposition 7 ([22]) *Pour SPM, pour tout $n \in \mathbb{N}^*$, la longueur du transitoire en mode parallèle $t_{par}(1)$ de la configuration (n) est comprise entre les bornes*

$$\mathcal{O}(n) = n + o(n) \leq t_{par}(1) \leq \left(1 + \frac{1}{2 + \sqrt{2}}\right) n + o(n) = \mathcal{O}(n) .$$

Piles de sable généralisées

Sommaire

2.1	Algorithme de calcul du point fixe	16
2.1.1	Découpage	16
2.1.2	Calcul	18
2.1.3	Fusion	21
2.2	Étude de l'algorithme	21
2.2.1	Exactitude	22
2.2.2	Complexité	23
2.3	Longueur du transitoire	24

Les résultats vus dans la partie précédente ne s'appliquent pas directement à des configurations de départ quelconques, seules les configurations à colonne unique ont été étudiées. Quelques tentatives de généralisation ont quand même été faites : par exemple dans [27] les auteurs remarquent qu'en partant de configurations décroissantes la structure de treillis est préservée pour SPM. Toujours pour ces mêmes configurations décroissantes et dans le cadre de SPM, l'ensemble des points fixes atteignables est caractérisé dans [43]. Cependant ces points fixes ne sont pas associés aux configurations qui les ont engendrées, et on ne sait rien des configurations encore plus générales.

Dans deux articles [24, 23], nous avons essayé de trouver une formule pour décrire le point fixe d'une configuration initiale quelconque. Cette partie reprend les résultats qui s'y trouvent ([24] pour SPM, [23] pour $\text{IPM}(k)$). On peut noter que le problème tel qu'il est posé ici n'a pas de solution évidente : il est en général impossible de prévoir en temps constant la structure précise du point fixe, les comportements possibles étant trop variés. Nous avons en revanche proposé un algorithme rapide qui, étant donnée une configuration initiale c , calcule son point fixe en temps $\mathcal{O}(l \cdot \min(l, n))$, où l est la longueur et n le nombre de grains de c .

Un des résultats majeurs de cette partie, outre l'algorithme optimisé, est le théorème 15 qui assure que pour tous les modèles $\text{IPM}(k)$, une configuration quelconque converge vers un point fixe qui reste unique. Par conséquent, le point fixe trouvé par notre algorithme est correct. Ce résultat était en partie seulement montré dans [28], pour les *Chip Firing Games* (CFG) de poids positif. SPM est un CFG de poids quelconque, alors que les $\text{IPM}(k)$ sont un modèle à part.

Nous présentons ici uniquement les résultats pour $\text{IPM}(k)$ qui se trouvent dans [23], et qui généralisent ceux de [24] pour SPM. À l'aide des définitions et notations introduites dans le cha-

pitre précédent, nous commençons par décrire l'algorithme, puis nous prouvons qu'il fonctionne de manière correcte et nous évaluons sa complexité.

2.1 Algorithme de calcul du point fixe

Pour calculer le point fixe d'une configuration c quelconque, nous proposons un algorithme, dont les différentes étapes sont illustrées sur la figure 2.1.

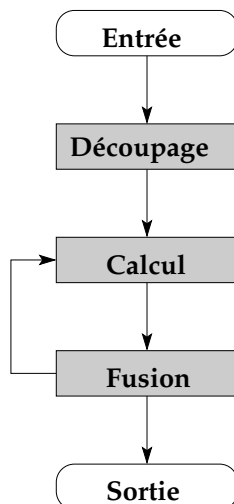


FIG. 8 – Structure de l'algorithme de calcul du point fixe.

L'algorithme débute par une phase d'initialisation, qui consiste à **découper** la configuration de départ en petits *intervalles* que l'on sait traiter. Ensuite on itère les deux phases suivantes : les **calculs** dans les intervalles, puis la **fusion** d'intervalles lorsque leur union permet de faire du calcul. Quand aucune fusion n'est plus possible, l'algorithme renvoie la configuration obtenue, qui est le point fixe désiré.

On remarque que ces différentes phases, et plus particulièrement l'itération des fusions, sont nécessaires. Par exemple pour une configuration $(m, 0, \dots, 0, n)$ à 2 colonnes isolées, il est impossible de savoir si l'effondrement de la première colonne de hauteur m va déborder sur le tas formé par la seconde colonne de hauteur n , ou rester bloqué contre. Tout cela dépend des valeurs de m , n , et de l'espace entre les deux. C'est à cause de ce phénomène que l'on ne peut pas prévoir le point fixe en temps constant.

Nous détaillons maintenant les différentes phases de l'algorithme.

2.1.1 Découpage

Le découpage de la configuration de départ se fait ainsi : chaque « morceau » de configuration (*intervalle*) doit être atteignable en partant d'une configuration à colonne unique, et doit être de longueur maximale tout en respectant cette propriété. C'est le découpage qui simplifie le problème initial, en permettant en même temps d'économiser du temps de calcul.

Même si cela semble intuitif, les résultats du chapitre précédent doivent être étendus. En effet, leurs auteurs supposaient que les configurations pouvaient croître autant que possible. Ici,

le mouvement est limité à des intervalles de longueur fixe, les grains ne peuvent pas se déplacer trop loin sur la droite.

Par la suite, on appelle *graphe des orbites bornées* la restriction de \mathcal{G}_c aux configurations de longueur au plus l , il est noté \mathcal{G}_c^l . On montre dans cette partie que $\mathcal{G}_{(n)}^l$ est toujours un treillis.

Lemme 8 *Soient un système $IPM(k)$, $k \in \mathbb{N}^*$, et une configuration c de longueur au plus $l' \leq l$ contenant n grains. Alors $c \in \mathcal{G}_{(n)}^l$ si et seulement si pour tous $1 \leq i < j \leq l'$, on a $j - i \leq k(c_i - c_j + 1)$.*

Preuve. Soit une configuration $c \in \mathcal{G}_{(n)}^l$, où n est le nombre de grains de c . Par définition, $\mathcal{G}_{(n)}^l$ est un sous-graphe de $\mathcal{G}_{(n)}$ donc $c \in \mathcal{G}_{(n)}$ et c satisfait les hypothèses du lemme 2 (page 11).

Inversement, soit c une configuration à n grains de longueur $l' \leq l$, telle que pour tous $1 \leq i < j \leq l'$ on ait $j - i \leq k(c_i - c_j + 1)$. D'après le lemme 2, $c \in \mathcal{G}_{(n)}$. Si l'on considère le chemin dans $\mathcal{G}_{(n)}$ qui remonte de c à la racine (n) du graphe, toutes les configurations sont de longueur inférieure à l car la règle locale ne peut pas diminuer la longueur d'une configuration. Toutes les applications de règles se font donc dans l'intervalle $[1, l - 1]$ et par conséquent ce chemin, ainsi que les configurations qui le composent, sont dans $\mathcal{G}_{(n)}^l$. \square

Les conditions d'appartenance formulées ainsi demandent un temps de vérification quadratique en la longueur, ce qui est trop élevé. Pour accélérer l'algorithme on utilise la formulation de la remarque 1 (page 11) qui se vérifie en temps linéaire : configuration décroissante, et entre 2 plateaux de longueur $k + 1$ il doit être possible de déplacer un grain (existence d'une falaise ou d'un plateau de longueur strictement inférieure à k entre les deux plateaux de longueur $k + 1$).

Proposition 9 *Pour tous les systèmes $IPM(k)$, pour tous $n, l \in \mathbb{N}^*$, $\mathcal{G}_{(n)}^l$ a un unique point fixe π^l .*

Preuve. Quels que soient $k, n, l \in \mathbb{N}^*$, d'après le lemme 8, un point fixe π^l quelconque de $\mathcal{G}_{(n)}^l$ est de longueur au plus l , a au plus un plateau de taille $k + 1$, deux plateaux de longueur inférieure ou égale à k aux extrémités et des plateaux de taille k partout ailleurs. Sa construction est représentée sur la figure 9 : c'est un escalier avec des marches de longueur k , les grains supplémentaires étant alors disposés de bas en haut (grains numérotés dans l'ordre sur la figure).

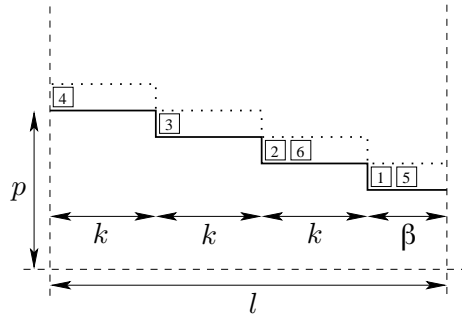


FIG. 9 – Structure du point fixe π^l pour $IPM(k)$.

Il est clair qu'une telle configuration est unique. Tout d'abord, la division entière de $l = \alpha k + \beta$, $\beta < k$ est unique. Soit f la fonction (croissante) comptant le nombre de grains dans l'escalier de longueur l , de hauteur p , définie par $f(p) = \frac{1}{2}k\alpha(2p - \alpha + 1) + \beta(p - \alpha)$. Il existe une seule

valeur de p telle que $f(p) \leq n < f(p+1)$, n étant le nombre de grains à répartir (sur la figure, $f(p)$ est l'escalier dessiné avec la ligne continue, $f(p+1)$ celui en pointillés). Les $x = n - f(p)$ grains restants sont répartis dans un premier temps sur les marches, tels que $x = \gamma(\alpha + 1) + y$ avec $\gamma \leq \beta$ maximal (décomposition unique). Pour finir, les y grains restants sont répartis sur les α marches restantes en $y = \delta\alpha + \lambda$, $\lambda < p - \alpha$ (décomposition unique). \square

La structure précise du point fixe est détaillée plus bas dans la proposition 11. Néanmoins, son unicité suffit à prouver le corollaire suivant.

Corollaire 10 *Pour tous les systèmes $IPM(k)$, pour tous $n, l \in \mathbb{N}^*$, $\mathcal{G}_{(n)}^l$ est un treillis fini.*

Preuve. D'après la proposition 9, deux configurations c^1 et c^2 de $\mathcal{G}_{(n)}^l$ atteignent le point fixe π . Il existe donc une borne supérieure (au plus (n)) et inférieure (au moins π) commune à c^1 et c^2 . De plus comme $\mathcal{G}_{(n)}^l$ est un sous-graphe de $\mathcal{G}_{(n)}$ (par définition de $\mathcal{G}_{(n)}^l$), il est fini et ne contient pas de cycle. \square

La phase de découpage peut avoir lieu en s'appuyant sur les résultats du lemme 8 et du corollaire 10. La configuration de départ est parcourue de gauche à droite en comptant le nombre de plateaux de taille $k+1$ rencontrés, un nouvel intervalle est créé dès qu'on en repère deux consécutifs. De cette manière on s'assure que chaque intervalle contient une configuration appartenant à un des $\mathcal{G}_{(n)}^l$, dont le point fixe est π^l . Le découpage est décrit par le pseudo-code 1.

Après découpage, les intervalles sont représentés par des intervalles d'entiers $I_i = [a_i, b_i]$ tels que $(c_j)_{j \in I_i} \in \mathcal{G}_{(n_i)}^{l_i}$ pour tout i , où $l_i = b_i - a_i + 1$ est la longueur de l'intervalle et $n_i = \sum_{j=a_i}^{b_i} c_j$ le nombre de grains qu'il contient. On remarque que l_i et n_i peuvent être calculés par la fonction de découpage sans augmentation de complexité.

La figure 10 représente un exemple de découpage pour $IPM(2)$.

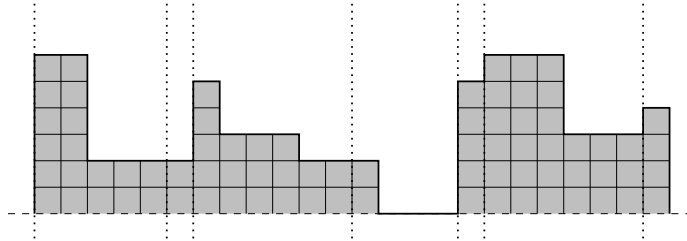


FIG. 10 – Exemple de découpage pour $IPM(2)$.

Remarque 3 *Dans la procédure de découpage, le dernier intervalle doit être traité différemment des autres puisqu'il est suivi de colonnes qui sont toutes à 0. On se place donc dans le cas de $IPM(k)$ à une colonne, en donnant comme longueur à l'intervalle une taille « infinie », i.e. la taille nécessaire à l'obtention du point fixe non borné. La valeur $l_i = \left\lceil \frac{k}{2}(\sqrt{8n_i/k+1} - 1) \right\rceil$, longueur du point fixe de $\mathcal{G}_{(n_i)}$ pour $IPM(k)$ obtenue d'après les résultats de [29] convient parfaitement.*

2.1.2 Calcul

Le but de cette phase est de donner l'expression du point fixe de chaque intervalle en temps constant. Tous ces points fixes mis bout à bout formeront une configuration atteignable depuis

```

procedure decoupe(c[], k) { // c est la configuration initiale
                                // k le modèle utilisé
    l = |c|; // longueur de c
    nbp = 0; // nombre de plateaux consécutifs
    lp = 1; // longueur du plateau en cours
    I = ∅; // ensemble des bornes supérieures des intervalles

    for (i=1; i<l; i++) {
        if (c[i+1] > c[i]) { // augmentation
            I = I ∪ {i};
            nbp = 0;
            lp = 1;
        } else if (c[i+1] < c[i]) {
            if (lp < k) || (c[i+1] - c[i] >= 2) { // mouvement possible
                nbp = 0;
            }
            lp = 1;
        } else if (c[i+1] == c[i]) { // plateau
            lp ++;
            if (lp == k+1) {
                nbp ++;
                if (nbp == 2) { // 2 plateaux consécutifs
                    I = I ∪ {i};
                    nbp = 0;
                }
                lp = k;
            }
        }
    }
    return I;
}
    
```

Pseudo-code 1 – Découpage de la configuration initiale en intervalles.

la configuration initiale, avec un gain de temps par rapport à la simulation naïve. Comme dans cette partie nous nous concentrons sur un seul intervalle, nous omettons les indices i .

Nous commençons par préciser le résultat de la proposition 9 en décrivant dans la proposition ci-dessous la structure de π^l .

Proposition 11 *Pour tous $k, n, l \in \mathbb{N}^*$, le point fixe π^l de $\mathcal{G}_{(n)}^l$ est défini par*

$$\pi^l = \left((p+1)^{[k]}, p^{[k]}, \dots, (p'+1)^{[k]}, p'^{[k+1]}, (p'-1)^{[k]}, \dots, (p-\alpha+\varepsilon+1)^{[k]}, (p-\alpha+\varepsilon)^{[k'']}] \right),$$

avec

$$\begin{aligned}
 \alpha &= \lfloor l/k \rfloor \\
 \beta &= l - k\alpha \\
 p &= \lfloor \frac{1}{l}(n + l\alpha - \frac{1}{2}k\alpha(\alpha+1)) \rfloor \\
 f(p) &= \frac{1}{2}k\alpha(2p - \alpha + 1) + \beta(p - \alpha)
 \end{aligned}$$

$$\begin{aligned}
 \gamma &= \min \left(\left\lfloor \frac{n-f(p)}{\alpha+1} \right\rfloor, \beta \right) \\
 \delta &= \begin{cases} \left\lfloor \frac{1}{\alpha}(n-f(p)-\gamma(\alpha+1)) \right\rfloor & \text{si } \gamma = \beta \\ 0 & \text{sinon} \end{cases} \\
 \lambda &= n - f(p) - \gamma(\alpha+1) - \delta\alpha \\
 k' &= \gamma + \delta \\
 \mathbb{1}_\lambda &= \begin{cases} 1 & \text{si } \lambda > 0 \\ 0 & \text{sinon} \end{cases} \\
 \varepsilon &= \begin{cases} 0 & \text{si } \gamma < \beta \\ 1 & \text{sinon} \end{cases} \\
 k'' &= \begin{cases} \beta - \gamma - \mathbb{1}_\lambda & \text{si } \gamma < \beta \\ k - \delta - \mathbb{1}_\lambda & \text{sinon} \end{cases} \\
 p' &= \begin{cases} (p - \alpha + \varepsilon) + \lambda & \text{si } \lambda > 0 \\ -1 & \text{sinon (i.e. } p' \text{ indéfini)} \end{cases} .
 \end{aligned}$$

Preuve. Comme vu dans la preuve de la proposition 9, le point fixe π^l est composé de plateaux de taille k partout sauf éventuellement un plateau de taille $k+1$, et deux plateaux de longueur inférieure à k aux extrémités (voir figure 9). Il peut donc s'exprimer sous la forme suivante :

$$\pi^l = \left((p+1)^{[k']}, p^{[k]}, \dots, (p'+1)^{[k]}, p'^{[k+1]}, (p'-1)^{[k]}, \dots, (p-\alpha+\varepsilon+1)^{[k]}, (p-\alpha+\varepsilon)^{[k'']} \right) .$$

Pour déterminer tous les paramètres, on a déjà vu que $l = \alpha k + \beta$, $\beta < k$, était la division entière de l par k . D'autre part, p est tel que $f(p) \leq n < f(p+1)$, d'où après résolution de l'équation $p = \lfloor \frac{1}{t}(n + l\alpha - \frac{1}{2}k\alpha(\alpha+1)) \rfloor$. On cherche ensuite à compléter avec les $n - f(p)$ grains restants les $\alpha+1$ marches une par une en partant du bas, jusqu'à avoir ajouté β grains sur chacune. Cela correspond à la division de $n - f(p)$ par $\alpha+1$, soit $\gamma = \min \left(\left\lfloor \frac{n-f(p)}{\alpha+1} \right\rfloor, \beta \right)$ grains par marche. Si la première marche est remplie, *i.e.* si $\gamma = \beta$, la marche la plus basse a pour hauteur $p - \alpha + 1$ soit $\varepsilon = 1$. Sinon, on a $\varepsilon = 0$. Si $\gamma = \beta$, les grains restants sont répartis sur les α marches qui restent, soit $\delta = \lfloor \frac{1}{\alpha}(n - f(p) - \gamma(\alpha+1)) \rfloor$ grains par marche. Il reste encore $\lambda = n - f(p) - \gamma(\alpha+1) - \delta\alpha$ grains supplémentaires qui déterminent où est le plateau de taille $k+1$ si $\lambda > 0$. On définit pour finir $\mathbb{1}_\lambda = 1$ si $\lambda > 0$, 0 sinon. Tous ces paramètres sont illustrés sur la figure 11.

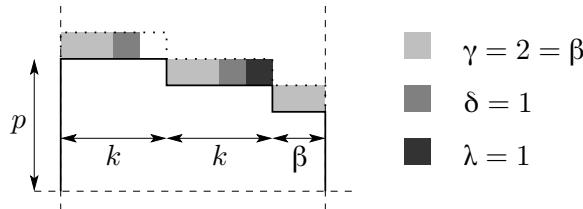


FIG. 11 – Illustration des paramètres définissant π^l .

À partir de ces premières valeurs, on déduit aisément qu'il y a $k' = \gamma + \delta$ grains à la hauteur $p+1$. De même, il y a $k'' = \beta - \gamma - \mathbb{1}_\lambda$ grains de hauteur la plus basse si $\gamma < \beta$, $k'' = k - \delta - \mathbb{1}_\lambda$

sinon. Enfin, la hauteur p' du plateau de taille $k + 1$ est celle du plateau le plus bas à laquelle on ajoute les λ grains supplémentaires, soit $p' = (p - \alpha + \varepsilon) + \lambda$. Si $\lambda = 0$, p' est indéfini et on peut lui affecter arbitrairement la valeur -1 . \square

Dans la formule de la proposition précédente il est possible que $k' = 0$ ou $k'' = 0$. On introduit donc deux nouvelles valeurs P et Q pour représenter les hauteurs des colonnes extrêmes de π^l . La hauteur de la colonne la plus à gauche est alors $P = p + 1$ si $k' \neq 0$, ou $P = p$ sinon. De même, celle de la colonne la plus à droite est $Q = p - \alpha - \varepsilon$ si $k'' \neq 0$, ou $Q = p - \alpha - \varepsilon + 1$ sinon.

Remarque 4 On remarque que dans le cas de SPM, cette expression est beaucoup plus simple car $\beta = 0$ et donc $\gamma = \delta = k' = 0$.

2.1.3 Fusion

La troisième partie de l'algorithme consiste à parcourir la configuration qui vient d'être obtenue après la phase de calcul, en mettant bout à bout tous les points fixes de chaque intervalle. On recherche des frontières pouvant être détruites tout en préservant la propriété d'appartenance à un graphe d'orbites bornées. Si l'on en trouve une, on fusionne les deux intervalles concernés et l'algorithme boucle sur l'étape de calcul. S'il n'y en a pas, l'algorithme termine et retourne la dernière configuration calculée.

Soit c^t la configuration sur laquelle travaille l'algorithme à l'itération t . Deux intervalles consécutifs I_i et I_{i+1} de c^t peuvent fusionner lorsqu'il y a une falaise sur la frontière, ou que la réunion des deux intervalles fait apparaître un plateau de longueur strictement inférieure à k . Formellement, si l'on décrit les deux morceaux de la configuration c^t par

$$\begin{aligned} (c_k^t)_{k \in I_i} &= (P_i^{[k_i^0]}, (P_i - 1)^{[k_i^1]}, \dots, Q_i^{[k_i^{w_i}]}) \\ (c_k^t)_{k \in I_{i+1}} &= (P_{i+1}^{[k_{i+1}^0]}, (P_{i+1} - 1)^{[k_{i+1}^1]}, \dots, Q_{i+1}^{[k_{i+1}^{w_{i+1}}]}) , \end{aligned}$$

alors les intervalles I_i et I_{i+1} fusionnent si l'une des conditions suivantes est vérifiée :

- (i) $Q_i \geq P_{i+1} + 2$ (falaise) ;
- (ii) $Q_i = P_{i+1} + 1$ et $(k_i^{w_i} < k$ ou $k_{i+1}^0 < k)$ (un des plateaux aux extrémités est de longueur inférieure à k) ;
- (iii) $Q_i = P_{i+1}$ et $k_i^{w_i} + k_{i+1}^0 < k$ (la réunion des plateaux reste de longueur inférieure à k).

En cas de fusion de deux intervalles, l'algorithme calcule le point fixe de leur réunion $I_i \cup I_{i+1}$. D'après le lemme 8, cette sous-configuration est bien dans $\mathcal{G}_{(n_i+n_{i+1})}^{l_i+l_{i+1}}$. On obtient ainsi une nouvelle configuration c^{t+1} sur laquelle une nouvelle fusion est recherchée. Dès que ce n'est plus possible, l'algorithme renvoie la configuration en cours. La partie suivante montre que c'est bien le point fixe que l'on recherche.

2.2 Étude de l'algorithme

Dans cette partie nous montrons que l'algorithme est correct en prouvant que le graphe des orbites d'une configuration quelconque est un treillis (théorème 15), et qu'il est suffisamment rapide pour justifier son utilisation (proposition 17).

2.2.1 Exactitude

Nous commençons par prouver que la configuration renvoyée par l'algorithme est un point fixe pour $\text{IPM}(k)$, et que tous les graphes \mathcal{G}_c avec c quelconque sont des treillis. Par conséquent, le point fixe retourné par l'algorithme est le seul possible.

Lemme 12 *Pour tous les systèmes $\text{IPM}(k)$, pour toute configuration initiale c , \mathcal{G}_c n'a pas de cycle.*

Preuve. Soit une configuration $c = (c_1, \dots, c_l)$ pour $\text{IPM}(k)$, et considérons la fonction $\varphi(c) = \sum_{i=1}^l \sum_{j=1}^{c_i} j$. Si $c' = (c'_1, \dots, c'_l)$ est obtenue en appliquant une règle à c (déplacement de la colonne i vers la colonne j , $1 \leq i < j \leq l$), alors $\varphi(c') = \varphi(c) - c_i + c'_j = \varphi(c) - c_i + c_j + 1$. Comme une règle était applicable, on a $c_i - c_j \geq 2$ donc $\varphi(c') < \varphi(c)$; en d'autres termes les règles de $\text{IPM}(k)$ sont irréversibles. \square

Proposition 13 *Pour toute configuration initiale, l'algorithme termine et retourne un point fixe.*

Preuve. À cause de l'irréversibilité (lemme 12) et du nombre fini de grains, l'algorithme termine.

Supposons qu'il ne renvoie pas un point fixe, *i.e.* qu'il retourne une configuration $c = (c_i)_{1 \leq i \leq l}$ pour laquelle il existe $1 \leq i < j \leq l$ tels qu'un grain peut aller de la colonne c_i vers la colonne c_j en appliquant une règle locale. Dans le cas où il s'agit de la règle verticale, il y a une falaise en c_i et $j = i + 1$. Celle-ci est nécessairement située à la frontière entre deux intervalles, à cause de la structure du point fixe sans falaise. Mais dans ce cas, les deux intervalles auraient dû fusionner et l'algorithme n'aurait pas terminé à cette itération.

De même s'il y a un plateau de longueur inférieure à k entre c_{i+1} et c_{j-1} , il est à cheval entre deux intervalles (éventuellement en « touchant » simplement l'un des deux) et donc les intervalles auraient dû fusionner. \square

Il ne reste plus qu'à démontrer le résultat principal de cette partie qui prouve que le point fixe est le seul possible. Nous nous servons pour cela du lemme technique suivant, qui prouve que tant qu'un mouvement est « bloqué » il en reste toujours un qui peut être effectué.

Lemme 14 *Soient un système $\text{IPM}(k)$ et une configuration c quelconque. Supposons qu'une règle de $\text{IPM}(k)$ est applicable dans c de la colonne i vers la colonne j , soit h tel que $i \leq h < j$. Tant qu'aucune règle n'est appliquée depuis une colonne i' vers une colonne j' telles que $i' \leq h < j'$, alors une règle de $\text{IPM}(k)$ reste toujours applicable.*

Preuve. Soit une configuration $c = (c_1, \dots, c_l)$ dans le modèle $\text{IPM}(k)$, $k \in \mathbb{N}^*$. Supposons tout d'abord que la règle verticale est applicable depuis la colonne i de c . La configuration c peut évoluer, mais tant qu'aucune règle n'est appliquée depuis une colonne $i' \leq i$ vers une colonne $j' > i$, c_i ne peut qu'augmenter et c_{i+1} ne peut que diminuer. Par conséquent la règle verticale pourra toujours être appliquée à la colonne i .

Supposons maintenant que la règle horizontale peut être appliquée dans c depuis la colonne i vers la colonne j , et qu'aucun déplacement de grain traversant la colonne i n'est effectué. Alors c_i ne peut qu'augmenter, tandis que pour tout h vérifiant $i < h \leq j$, c_h ne peut que diminuer. Après un nombre quelconque d'étapes, on obtient une configuration c' vérifiant l'un des cas suivants.

- (i) Soit aucun des c_h , $i \leq h \leq j$, n'a changé, la règle horizontale peut toujours être appliquée de i vers j dans c' .

- (ii) Soit c_i augmente, alors $c'_i > c_i = c_{i+1} + 1 \geq c'_{i+1} + 1$, la règle verticale est applicable en i dans c' .
- (iii) Soit c_i est inchangé et c_{i+1} diminue, la règle verticale est aussi applicable en i dans c' .
- (iv) Soit c_i et c_{i+1} sont inchangés, il existe $i + 1 < h \leq j$ tel que $c'_h < c_h$. Prenons le plus petit $h > i + 1$ vérifiant $c'_h < c_h$.
 - Si $h = j$, alors $c'_{j-1} = c_{j-1} = c_j + 1 > c'_j + 1$, la règle verticale peut être appliquée en $j - 1$ dans c' .
 - Si $h < j$ et $c'_h = c_h - 1$, la règle horizontale peut déplacer un grain de c' de la colonne i à la colonne h .
 - Si $h < j$ et $c'_h \leq c_h - 2$, on peut appliquer la règle verticale à c' à la position $h - 1$. \square

Théorème 15 *Pour tous les systèmes $IPM(k)$, pour toute configuration c , \mathcal{G}_c est un treillis.*

Preuve. Soit une configuration c et un modèle $IPM(k)$. Notons I_i^f les derniers intervalles obtenus après application de l'algorithme à c , juste avant qu'il ne termine. Découpons c selon ces intervalles (différents de ceux calculés par la première phase de découpage) en sous-configurations que l'on note $c_{[i]} = (c_j)_{j \in I_i^f}$.

Aucun grain d'aucun $c_{[i]}$ ne se déplace jamais vers un autre $c_{[j]}$. En effet, supposons que ce n'est pas le cas et que de tels i et j existent ; nous exhibons une orbite de c qui mène à contradiction. Simulons l'évolution de c en n'effectuant aucun déplacement de grain depuis une colonne de $c_{[i]}$ vers une colonne de $c_{[j]}$. Au bout d'un moment, toutes les configurations partielles $c_{[h]}$ aboutissent à des points fixes partiels que l'on nomme $d_{[h]}$. Chacun de ces $d_{[h]}$ correspond aux $c_{[h]}^f$ calculés par l'algorithme car ils appartiennent au même graphe des orbites bornées, qui est un treillis d'après le corollaire 10, et ils ne reçoivent ni ne perdent aucun grain par hypothèse. Par conséquent, la configuration d obtenue en mettant bout à bout les $d_{[h]}$ est le point fixe c^f trouvé par l'algorithme. Cependant d'après le lemme 14, comme il y a eu un mouvement possible depuis un intervalle de c vers un autre et qu'aucun n'a été effectué, d ne peut être un point fixe.

Par conséquent, le comportement des $c_{[h]}$ n'est pas influencé par les autres. Le graphe des orbites de c est donc le produit des graphes des orbites bornées des $c_{[h]}$ et est un treillis en tant que produit de treillis. \square

Corollaire 16 *Pour tous les systèmes $IPM(k)$, pour toute configuration c , le point fixe de \mathcal{G}_c coïncide avec le point fixe trouvé par l'algorithme.*

2.2.2 Complexité

La simulation naïve d' $IPM(k)$ sur une configuration c de longueur l à n grains se fait en temps $\mathcal{O}(l \cdot n^{3/2})$ (l pour détecter un endroit où appliquer la règle, et $n^{3/2}$ applications au maximum). Notre algorithme a une complexité en $\mathcal{O}(l \cdot \min(l, n))$ comme le montre la proposition suivante, soit un gain dans le pire des cas de $\mathcal{O}(n^{1/2})$.

Proposition 17 *Pour tous les systèmes $IPM(k)$, pour toute configuration c de longueur l à n grains, l'algorithme calcule le point fixe de c en temps $\mathcal{O}(l \cdot \min(l, n))$.*

Preuve. Soient un système $IPM(k)$, $k \in \mathbb{N}^*$, et une configuration c à n grains, de longueur l . Comme vu dans la partie 2.1.1, la complexité du **découpage** de c est en $\mathcal{O}(l)$ (fonction 1). Les

calculs se font en temps constant (proposition 11) pour chaque intervalle. Ils sont effectués l fois au plus lors du premier passage, puis une fois à chaque fusion. L'étape de **fusion** est en $\mathcal{O}(l)$ (parcours des intervalles, moins de l intervalles).

On remarque pour finir qu'il n'y a pas plus de $\min(n, l)$ itérations. Il ne peut y avoir plus de fusions que d'intervalles, c'est donc inférieur à l . De plus, chaque intervalle qui fusionne doit contenir au moins un grain qui ne bouge pas : prenons n'importe quel grain à la hauteur 0, il ne se déplace jamais. Il n'y a donc pas plus de fusions que de grains, soit moins de n itérations.

On retrouve bien la complexité finale en $\mathcal{O}(l + l + \min(l, n) \cdot (l + 1)) = \mathcal{O}(l \cdot \min(l, n))$. \square

Cette borne peut être atteinte, comme le montre l'exemple suivant.

Exemple 1 Étant donné un modèle $IPM(k)$ et $n = 7m$ un multiple de 7, soit la configuration c^n définie par

$$c_i^n = \begin{cases} 7 & \text{si } i = (k+3)j, 0 \leq j < (k+3)\frac{n}{7} \\ 0 & \text{sinon.} \end{cases}$$

En d'autres termes, c^n est constituée de $n/7$ colonnes de hauteur 7 séparées par $k+2$ colonnes à 0. Elle a donc n grains et est de longueur $(k+3)(\frac{n}{7} - 1) + 1 = \mathcal{O}(n)$. Elle est découpée en $n/7$ intervalles $(7, \underbrace{0, \dots, 0}_{k+1}, 0)$ et $n/7 - 1$ intervalles (0) (voir figure 12).

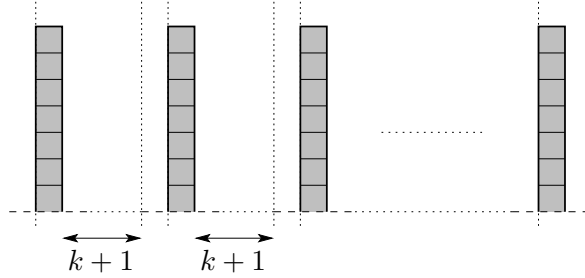


FIG. 12 – Exemple de configuration donnant lieu à $\mathcal{O}(n) = \mathcal{O}(l)$ fusions.

Quelle que soit la valeur de k , le point fixe de l'intervalle $(7, 0, \dots, 0)$ peut fusionner avec l'intervalle (0) qui suit. Il y aura donc $n/7$ fusions.

En général, on reste très en dessous de la borne supérieure donnée par la proposition 17. L'algorithme est très rapide par exemple sur les configurations éparpillées où $l \gg n$ (cas extrême, aucune fusion et donc temps en $\mathcal{O}(l)$). Il l'est encore plus sur les configurations denses où $n \gg l$, notamment lorsque $l = 1$, il fonctionne en temps constant (un seul intervalle).

2.3 Longueur du transitoire

Dans le cas de SPM, on peut ajouter à la phase de **calcul** le calcul de la longueur du transitoire en mode *séquentiel*. En effet il est possible de calculer le temps nécessaire t_i pour obtenir le point fixe π^{l_i} du i^e intervalle (en mode séquentiel). Si $k = 1$, on a

$$\begin{aligned} t_i &= \sum_{j=0}^{l_i-1} j(p_i - j) + \sum_{j=p_i-l_i-\lambda_i+1}^{p_i-l_i+1} j - t_i^0 \\ &= \frac{1}{6}l_i(l_i-1)(3p_i-2l_i+1) + (p_i-l_i-\frac{1}{2}\lambda_i+1)(\lambda_i+1) - t_i^0 \end{aligned}$$

en ajoutant les temps individuels des grains de l'escalier de hauteur maximale p_i , ceux des λ_i grains supplémentaires et en retranchant les $t_i^0 = \sum_{j=0}^{l_i-1} j \cdot c_{j+m}$ pas qui ont déjà été effectués ($m = \inf I_i$).

On commence par faire la somme de tous ces temps partiels $t = \sum_i t_i$, puis à chaque fusion d'intervalles i et $i+1$, le nouveau temps t' s'exprime par

$$\begin{aligned} t' &= t + \frac{1}{6}l'_i(l'_i - 1)(3p'_i - 2l'_i + 1) + (p'_i - l'_i - \frac{1}{2}\lambda'_i + 1)(\lambda'_i + 1) \\ &- t_i - t_{i+1} + n_{i+1} \cdot l_i \end{aligned}$$

où p'_i est la nouvelle valeur de p_i , l'_i celle de l_i et λ'_i celle de λ_i . Le dernier terme correspond au déplacement (qui n'a pas eu lieu mais qui est comptabilisé dans les premiers termes) des n_{i+1} grains de I_{i+1} de l_i cases sur la droite.

Remarque 5 *Un tel calcul n'a pas de sens pour les modèles $IPM(k)$, car tous les chemins n'ont pas la même longueur (proposition 6, page 12). Comme notre algorithme en privilégie un, qui n'est pas forcément le plus court ni le plus long, calculer sa longueur ne servirait à rien.*

Piles de sable symétriques

Sommaire

3.1	Définition de SSPM	28
3.2	Graphe des orbites	30
3.3	Points fixes	35
3.4	Longueur du transitoire	38
3.4.1	Longueur minimale du transitoire	38
3.4.2	À propos de la longueur maximale du transitoire	41
3.5	Perspectives	42

L'élément qui est peut-être le plus étonnant dans les modèles SPM et $IPM(k)$ tels qu'ils sont introduits dans le chapitre 1 d'après [27, 29], est le fait que les grains de sable ne peuvent tomber que d'un côté. C'est un manque de réalisme évident, une pile de particules quelconques n'ayant aucune raison de privilégier un côté plutôt qu'un autre. On pourrait évidemment contourner ce problème en exécutant par exemple des simulations qui fonctionnent comme SPM pour la moitié des grains, et comme son symétrique pour l'autre moitié, en recollant les morceaux ensuite. Mais il manquerait un certain formalisme et il serait difficile d'affirmer que le modèle est proche de la réalité.

Très peu de résultats existent concernant ce problème bien connu. C'est principalement dû au fait que le graphe des orbites d'un tel modèle ne peut pas être un treillis : intuitivement, on sent bien que le point fixe n'est pas nécessairement le même si l'on commence à faire tomber les grains sur la droite puis sur la gauche, ou si l'on fait l'inverse.

Nous proposons dans cette partie un modèle symétrique (SSPM, *Symmetric Sand Pile Model* défini et étudié dans [26] puis [25]), basé sur la règle locale de SPM et sur cette idée que les grains peuvent se déplacer indifféremment à droite ou à gauche. Cela correspond à une manière de formaliser le comportement décrit plus haut, mais en considérant la configuration dans son ensemble et pas une moitié après l'autre.

Les résultats obtenus sont extrêmement intéressants, bien que la structure du graphe des orbites ne soit pas un treillis. En effet nous avons pu caractériser très précisément dans [26] les configurations appartenant au graphe des orbites d'une colonne unique (propositions 26 et 28), ce qui nous a permis de décrire ses points fixes potentiels et de les compter : quel que soit le nombre n de grains initial, il y a exactement $\lfloor \sqrt{n} \rfloor$ points fixes possibles (proposition 32). Tous ces résultats sont basés sur le fait qu'une pile de sable atteignable doit être symétrique (une partie croissante et une partie décroissante), en particulier les points fixes sont des « pyramides »

régulières (voir figures 16 et annexe A pour des exemples), ce qui tend à valider le côté réaliste de notre modèle. Nous prolongeons l'étude de SSPM dans [25] en montrant que la longueur du transitoire est au moins en $\mathcal{O}(n^{3/2})$ (autant que pour SPM) et au plus en $\mathcal{O}(n^2)$ selon le chemin choisi. Il semble que cette borne supérieure est exagérée, en nous appuyant sur des expérimentations, nous conjecturons que la longueur maximale du transitoire est elle aussi en $\mathcal{O}(n^{3/2})$.

Nous pensons que ce modèle devrait ouvrir la voie à l'étude de nouveaux modèles plus généraux que SPM et IPM(k). Il est possible de déduire du modèle SSPM un modèle SIPM(k) (*Symmetric Ice Pile Model*) ayant des propriétés similaires, puis d'étendre ces modèles aux dimensions supérieures. Pour atteindre un niveau de réalisme idéal, il ne resterait plus alors qu'à comprendre ce qu'il se passe à partir d'une configuration quelconque, et surtout trouver un modèle qui fonctionne « correctement » en mode parallèle, car le nôtre ne peut être défini que de manière séquentielle.

3.1 Définition de SSPM

Nous nous basons sur les définitions et les notations introduites dans le chapitre 1. Pour rappel, une *pile de sable* ou *configuration* est une suite finie d'entiers positifs $c = (c_1, \dots, c_k) \in (\mathbb{N}^*)^k$ où k est la *longueur* de la configuration. L'ensemble des configurations $\cup_{k \in \mathbb{N}} (\mathbb{N}^*)^k$ est noté \mathfrak{C} .

Le modèle SPM est étendu de la manière suivante : s'il y a une falaise sur la gauche alors un grain peut tomber à gauche, de même s'il y a une falaise sur la droite un grain peut tomber à droite. Commençons par définir les deux règles locales (règles *verticales* à gauche et à droite) :

$$\begin{aligned} V_i^l(c_1, \dots, c_k) &= \begin{cases} (c_1, \dots, c_{i-1} + \mathbf{1}, c_i - \mathbf{1}, \dots, c_k) & \text{si } i \neq 1, \\ (\mathbf{1}, c_1 - \mathbf{1}, \dots, c_k) & \text{sinon,} \end{cases} \\ V_i^r(c_1, \dots, c_k) &= \begin{cases} (c_1, \dots, c_i - \mathbf{1}, c_{i+1} + \mathbf{1}, \dots, c_k) & \text{si } i \neq k, \\ (c_1, \dots, c_k - \mathbf{1}, \mathbf{1}) & \text{sinon.} \end{cases} \end{aligned}$$

Ces règles déplacent un grain respectivement vers la gauche ou vers la droite. Notons $\delta_i^l(c) = c_i - c_{i-1}$ la *différence* en nombre de grains entre la colonne i et la colonne $i - 1$, en définissant $\delta_1^l(c) = c_1$. De même, soit $\delta_i^r(c) = c_i - c_{i+1}$ la différence entre la colonne i et la colonne $i + 1$, avec $\delta_k^r(c) = c_k$. Lorsque l'un de ces δ_i est supérieur ou égal à 2, la règle verticale correspondante peut être appliquée en i . La fonction « étape suivante » $f : \mathfrak{C} \mapsto \mathfrak{P}(\mathfrak{C})$ associe à une configuration l'ensemble de ses successeurs possibles (en mode *séquentiel*), soit

$$\forall c \in \mathfrak{C}, \bar{f}(c) = \{V_i^l(c) \mid \delta_i^l \geq 2, 1 \leq i \leq k\} \cup \{V_i^r(c) \mid \delta_i^r \geq 2, 1 \leq i \leq k\} .$$

La *règle globale* de SSPM est alors la fonction $f : \mathfrak{P}(\mathfrak{C}) \mapsto \mathfrak{P}(\mathfrak{C})$ qui caractérise l'évolution du système du temps t au temps $t + 1$:

$$\forall S \in \mathfrak{P}(\mathfrak{C}), f(S) = \bigcup_{c \in S} \bar{f}(c) .$$

Notons qu'avec cette définition, les transitions sont faites séquentiellement et non pas en parallèle. Si aucune règle locale ne peut être appliquée sur une configuration c , *i.e.* si $f(\{c\}) = \emptyset$, c est dite (par abus de terminologie) *point fixe* pour SSPM.

Le *graphe des orbites* peut être défini naturellement, en reliant deux configurations c et c' lorsque $c' \in f(\{c\})$. Avant d'examiner en détail les configurations appartenant au graphe des

orbites dans la section suivante, nous commençons par décrire sa structure. Avec l'aide d'une énergie potentielle, nous montrons qu'il est fini et sans cycle, c'est-à-dire que quelle que soit la configuration initiale un point fixe est atteint.

L'énergie d'une configuration $c = (c_1, \dots, c_k)$ est la somme des hauteurs de chacun de ses grains,

$$\bar{E}(c) = \sum_{i=1}^k \sum_{j=1}^{c_i} j .$$

Cette définition peut être étendue aux ensembles de configurations ainsi :

$$\forall S \in \mathfrak{P}(\mathfrak{C}), E(S) = \max_{c \in S} E(c) ,$$

avec $E(\emptyset) = 0$. Les deux lemmes qui suivent explicitent les propriétés basiques de cette énergie.

Lemme 18 *Pour toute configuration c à n grains, $\bar{E}(c) \leq \bar{E}((n))$ avec égalité si et seulement si $c = (n)$.*

Preuve. Considérons une configuration $c = (c_1, \dots, c_k)$ à n grains, soit $n = \sum_{i=1}^k c_i$. Posons $h(i) = \sum_{j=c_1+\dots+c_{i-1}+1}^{c_1+\dots+c_i} j$. Alors $\bar{E}((n)) = \sum_{j=1}^n j$ peut être réécrit en $\bar{E}((n)) = \sum_{j=1}^{c_1} j + \sum_{j=c_1+1}^{c_1+c_2} j + \dots + \sum_{j=c_1+\dots+c_{k-1}+1}^{c_1+\dots+c_k} j = \sum_{i=1}^k h(i)$. Comme $h(i) \geq \sum_{j=1}^{c_i} j$ pour tous $1 \leq i \leq k$ avec égalité si et seulement si $i = 1$, on a le résultat. \square

Lemme 19 *Pour tout ensemble de configurations $S \in \mathfrak{P}(\mathfrak{C})$ non vide, $E(f(S)) < E(S)$.*

Preuve. D'après la définition des règles V_i et de \bar{f} , il est évident que pour tout $c' \in \bar{f}(c)$ on a $\bar{E}(c') \leq \bar{E}(c) - 1$, d'où le résultat. \square

La proposition suivante permet maintenant de donner la structure globale du graphe des orbites d'une configuration quelconque.

Proposition 20 *Pour toute configuration initiale c , \mathcal{G}_c est fini et sans cycle, et donc contient au moins un point fixe.*

Preuve. Soit une configuration c quelconque, son énergie est nécessairement finie. D'après le lemme 19, $E(\{c\}) > E(f(\{c\})) > \dots > E(f^t(\{c\})) > \dots > 0$. Comme E est une fonction entière et positive, il existe nécessairement $u \in \mathbb{N}$ tel que pour tout $v \geq u$, $f^v(\{c\}) = \emptyset$. Donc $f^{u-1}(\{c\})$ contient un point fixe, et de plus le graphe \mathcal{G}_c est fini car $|f^t(\{c\})|$ est fini pour tout $0 \leq t \leq u$. Finalement, il n'y a pas de cycle dans \mathcal{G}_c car sinon les éléments de ce cycle contrediraient le lemme 19. \square

Corollaire 21 *SSPM a une dynamique de type point fixe.*

Ce corollaire permet de dire, comme c'était le cas pour SPM et IPM(k), qu'indépendamment des règles utilisées on atteint un point fixe. La différence majeure est que ces points fixes peuvent être multiples, comme le montre la figure 13 représentant le graphe des orbites de (5) pour SSPM. Cela vient du fait qu'en plus du non-déterminisme dû au choix de l'endroit où appliquer une règle, dans SSPM il faut également choisir arbitrairement quelle règle appliquer à une même colonne. Malgré tout, ce n'est pas gênant pour caractériser précisément les points fixes à partir de la caractérisation des éléments du graphe.

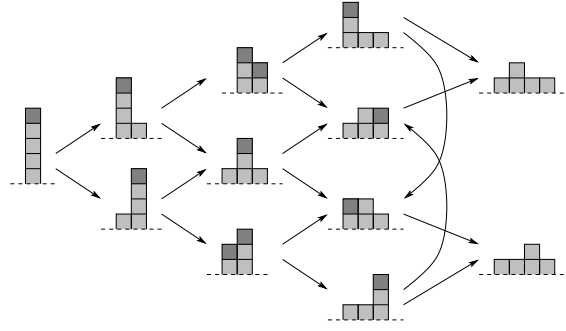


FIG. 13 – Graphe des orbites $\mathcal{G}_{(5)}$ d'une colonne unique à 5 grains pour SSPM. Seuls les grains en gris foncé sont susceptibles de tomber. On remarque qu'il y a deux points fixes différents.

Remarque 6 Deux configurations sont considérées comme identiques si elles ont la « même forme ». Cela ne tient pas compte des décalages éventuels qui pourraient se produire mais qui ne nous intéressent pas pour notre étude. En gardant en mémoire les positions des colonnes, on obtiendrait plusieurs fois la même configuration à des indices différents, ce qui reviendrait à considérer les classes d'équivalence des configurations (à décalage près).

3.2 Graphe des orbites

Nous étudions ici uniquement le cas où la configuration initiale est restreinte à une seule colonne, *i.e.* $c = (n)$ où n est le nombre de grains de la pile. Nous effectuons un travail semblable à celui qui a été effectué dans [29] pour SPM et IPM(k), c'est-à-dire caractériser les éléments du graphe des orbites $\mathcal{G}_{(n)}$ à l'aide des propositions 26 et 28.

Nous avons vu précédemment (proposition 20) que le graphe des orbites était fini et contenait au moins un point fixe, mais que celui-ci n'était pas nécessairement unique. Ce n'est pas un treillis, mais il est toujours possible de caractériser ses éléments. Pour cela nous disons qu'une configuration c de longueur k est *LR-décomposable* si elle peut être divisée en deux régions $L(c) = [1, t]$ (partie gauche de c) et $R(c) = [t + 1, k]$ (partie droite de c) telles que

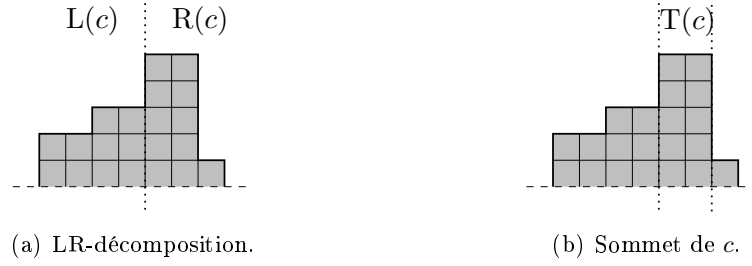
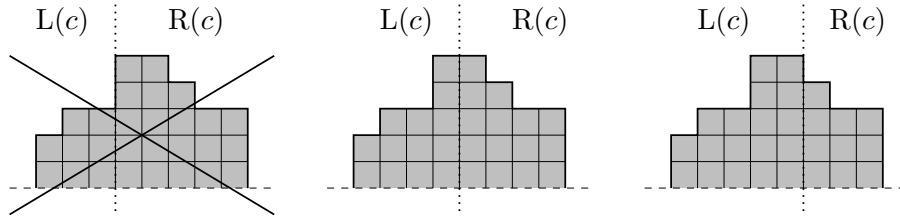
- (i) $\forall i \in L(c), i \neq t, c_i \leq c_{i+1}$, *i.e.* $L(c)$ est croissante ;
- (ii) $\forall i \in R(c), i \neq k, c_i \geq c_{i+1}$, *i.e.* $R(c)$ est décroissante.

On dit alors que $(L(c), R(c))$ est une *LR-décomposition* de c . La figure 14(a) montre un exemple de LR-décomposition. De plus, pour toute configuration c , le *sommet* de c est l'ensemble $T(c) = \{i \in [1, k] \mid \forall j \in [1, k], c_i \geq c_j\}$ (figure 14(b)).

Pour finir, pour une configuration $c = (c_1, \dots, c_k)$, un ensemble d'indices consécutifs $I = [\alpha, \beta] \subseteq [1, k]$ est dit *cassé* lorsque deux plateaux de $(c_\alpha, \dots, c_\beta)$ sont séparés par au moins une falaise. On utilise ici la seconde formulation du lemme 2 (page 11), équivalente ici au fait que pour tous $\alpha \leq i < j \leq \beta$, on a $j - i \leq c_i - c_j + 1$. La configuration c a une *LR-décomposition cassée* si elle admet une LR-décomposition $(L(c), R(c))$ telle que $L(c)$ et $R(c)$ soient cassés (voir figure 15).

Comme on peut le voir sur la figure 15, une configuration peut avoir plusieurs LR-décompositions valides. Les propositions suivantes permettent d'isoler celles qui nous intéressent, leurs preuves sont effectuées progressivement à l'aide de plusieurs lemmes techniques.

Lemme 22 Pour tout $n \in \mathbb{N}^*$ et $c \in \mathcal{G}_{(n)}$, c est LR-décomposable.


 FIG. 14 – Décomposition d'une configuration c .

 FIG. 15 – Exemples de LR-décompositions d'un configuration c . Les deux dernières sont cassées, pas la première.

Preuve. L'énoncé est vrai pour la configuration initiale (n) . Par récurrence, supposons que $c = (c_1, \dots, c_k) \in \mathcal{G}_{(n)}$ est LR-décomposable en $(L(c) = [1, t], R(c) = [t + 1, k])$. Soit $d \in \bar{f}(c)$, supposons qu'il existe i tel que $d = V_i^r(c)$, trois cas se présentent :

- (i) $i = t$; alors $L(d) = L(c) \setminus \{t\}$ et $R(d) = R(c) \cup \{t\}$. $L(d)$ est croissant car inclus dans $L(c)$, $R(d)$ est décroissant car $d_t \geq d_{t+1}$ et $R(c) \subseteq R(d)$;
- (ii) $i \in [t + 1, k - 1]$; alors $L(d) = L(c)$ et $R(d) = R(c)$. On remarque que $L(d)$ est croissant et $R(d)$ décroissant ;
- (iii) $i = k$; alors $L(d) = L(c)$ et $R(d) = R(c) \cup \{k + 1\}$. $L(d)$ est croissant, $R(d)$ est décroissant puisque $R(c) \subseteq R(d)$ et $d_k \geq 1$.

La preuve est semblable s'il existe i tel que $d = V_i^l(c)$. □

Lemme 23 Pour tout $n \in \mathbb{N}^*$ et $c \in \mathcal{G}_{(n)}$ de sommet $T(c)$, toute LR-décomposition de c est telle que $L(c) \setminus T(c)$ et $R(c) \setminus T(c)$ n'ont pas de plateaux de longueur strictement supérieure à 2.

Preuve. Soit $n \in \mathbb{N}^*$ et $c \in \mathcal{G}_{(n)}$. Remarquons que si $c = (n)$ l'énoncé est vérifié. Sinon si $c \neq (n)$, alors c a au moins un antécédent dans $\mathcal{G}_{(n)}$. On montre que l'énoncé est vrai pour $R(c) \setminus T(c)$ par l'absurde. Supposons qu'il y ait un plateau de longueur $m > 2$ dans $R(c) \setminus T(c)$, i.e. il existe $i \in R(c) \setminus T(c)$ tel que $c_i = c_{i+1} = \dots = c_{i+m-1}$. Par hypothèse, $c_{i-1} > c_i$ et $c_{i+m-1} > c_{i+m}$ (avec $c_{k+1} = 0$).

Soit une configuration d qui crée ce plateau par une avalanche sur la droite, c'est-à-dire telle que $c \in \bar{f}(d)$ et $c = V_j^r(d)$ pour un certain $j \in [i - 1, i + m - 1]$. Alors d n'est pas LR-décomposable et, d'après le lemme 22, n'est pas dans $\mathcal{G}_{(n)}$. De même une configuration d telle que $c = V_j^l(d)$ pour un certain $j \in [i, i + m]$ n'est pas LR-décomposable. Si l'on parcourt le graphe en remontant

de c jusqu'à la première avalanche qui crée le plateau, on obtient une configuration qui ne devrait pas y appartenir, d'où la contradiction. La preuve pour $L(c) \setminus T(c)$ est similaire. \square

Lemme 24 *Pour tout $n \in \mathbb{N}^*$ et $c \in \mathcal{G}_{(n)}$ de sommet $T(c)$, toute LR-décomposition de c est telle que $L(c) \setminus T(c)$ et $R(c) \setminus T(c)$ sont cassés.*

Preuve. Soit c une configuration pour laquelle il existe une LR-décomposition telle que $R(c) \setminus T(c) = [t, k]$ ne soit pas cassé, i.e. $R(c) \setminus T(c)$ contient deux plateaux non séparés par une falaise. Il y a donc deux indices $i, j \in [t, k]$, $i < j$, tels que $c_{i-1} > c_i = c_{i+1}$, $c_j = c_{j+1} > c_{j+2}$ et pour tout $h \in [i+1, j-1]$, $c_h = c_{h+1} + 1$. Soit $\delta = j - i$, on prouve par induction sur δ que $c \notin \mathcal{G}_{(n)}$. Si $\delta = 1$, le lemme 23 prouve le résultat. Supposons qu'il est vrai pour tout $\delta \in [1, m]$, posons $\delta = m + 1$. Soit une configuration c contenant deux plateaux non séparés par une falaise, à distance $j - i = m + 1$. Soit un antécédent $d \in \mathcal{G}_{(n)}$ de c tel que $c \in \bar{f}(d)$ et $c = V_h^r(d)$ pour $h \in [i-1, j+1]$. On obtient l'un des cas suivants pour h :

- (i) $h = i - 1$; alors $d_{i-1} > d_i < d_{i+1}$, d n'est pas LR-décomposable et $d \notin \mathcal{G}_{(n)}$ d'après le lemme 22 ;
- (ii) $h \in [i, j - 2]$; alors il y a un plateau dans d à la position $h + 1$ et le plateau en j est inchangé, donc par induction sur δ on a $d \notin \mathcal{G}_{(n)}$;
- (iii) $h = j - 1$; alors $d_{j-1} > d_j < d_{j+1}$, d n'est pas LR-décomposable et $d \notin \mathcal{G}_{(n)}$ par le lemme 22 ;
- (iv) $h = j$; alors il y a deux plateaux dans d aux positions $j - 1$ et i , $d \notin \mathcal{G}_{(n)}$ par induction sur δ ;
- (v) $h = j + 1$; alors $d_{j-1} > d_j < d_{j+1}$, d n'est pas LR-décomposable et $d \notin \mathcal{G}_{(n)}$ (lemme 22).

Si d est tel que $c = V_h^l(d)$ pour $h \in [i, j + 2]$, les seules valeurs possibles pour h sont les parties non strictement décroissantes, soit $h \in \{i + 1, j + 1\}$:

- (i) si $h = i + 1$; il y a deux plateaux dans d aux positions $i + 1$ et j donc par induction sur δ on a $d \notin \mathcal{G}_{(n)}$;
- (ii) si $h = j + 1$; $d_{j-1} > d_j < d_{j+1}$ et $d \notin \mathcal{G}_{(n)}$ d'après le lemme 22.

Donc parmi tous les antécédents de c qui peuvent créer l'un des plateaux, aucun ne peut appartenir à $\mathcal{G}_{(n)}$, ce qui contredit l'hypothèse de départ et $R(c) \setminus T(c)$ est cassé. Une preuve semblable peut être effectuée pour $L(c) \setminus T(c)$. \square

À partir d'exemples simples (voir annexe A), on observe que $T(c)$ peut contenir 1, 2, 3 ou 4 éléments. Ce dernier lemme prouve que pour toute configuration d'un graphe des orbites, ce sont les seules valeurs possibles.

Lemme 25 *Pour tout $n \in \mathbb{N}^*$ et $c \in \mathcal{G}_{(n)}$ de sommet $T(c)$, $|T(c)| \leq 4$.*

Preuve. Soit $c \in \mathcal{G}_{(n)}$, si $c = (n)$ alors l'énoncé est vérifié. Sinon, c a un antécédent dans $\mathcal{G}_{(n)}$. Supposons par l'absurde que $|T(c)| > 4$, et notons $T(c) = [\alpha, \beta]$.

Soit une configuration LR-décomposable d qui crée ce plateau, donc telle que $c = V_i^l(d)$ pour $i \in T(c)$. Pour respecter le lemme 22, d ne peut avoir que 2 formes :

- (i) $\dots, c_{\alpha-1} - 1, c_{\alpha} + 1, c_{\alpha+1}, c_{\alpha+2}, c_{\alpha+3}, \dots$ avec $c_{\alpha} + 1 > c_{\alpha+1} = c_{\alpha+2} = c_{\alpha+3}$;
- (ii) $\dots, c_{\alpha} - 1, c_{\alpha+1} + 1, c_{\alpha+2}, c_{\alpha+3}, c_{\alpha+4}, \dots$ avec $c_{\alpha+1} + 1 > c_{\alpha+2} = c_{\alpha+3} = c_{\alpha+4}$;

À la fois (i) et (ii) contredisent le lemme 23, et il en est de même pour tout d tel que $c = V_i^r(d)$ pour $i \in T(c)$. \square

La proposition principale peut maintenant être prouvée, elle est technique car beaucoup de cas doivent être pris en compte mais chacun d'eux est résolu simplement à l'aide de l'un des lemmes précédents.

Proposition 26 *Pour tout $n \in \mathbb{N}^*$ et $c \in \mathcal{G}_{(n)}$, c admet une LR-décomposition cassée.*

Preuve. Soient $n \in \mathbb{N}^*$ et $c = (c_1, \dots, c_k) \in \mathcal{G}_{(n)}$ de sommet $T(c)$. On distingue 4 cas selon la cardinalité de $T(c)$ (lemme 25).

- Si $|T(c)| = 1$, alors d'après le lemme 24 n'importe quelle LR-décomposition de c est cassée, car il n'y a aucun plateau supplémentaire dans $T(c)$.
- Si $|T(c)| = 2$, posons $L(c) = [1, t]$ et $R(c) = [t + 1, k]$ avec $T(c) = [t, t + 1]$. Puisque c est LR-décomposable (lemme 22), cette LR-décomposition est valide. Là non plus il n'y a pas de plateaux dans $L(c) \cap T(c)$ et dans $R(c) \cap T(c)$, donc c'est une LR-décomposition cassée (lemme 24).
- Si $|T(c)| = 4$, soit $t \in [1, k]$ tel que $T(c) = \{t, t + 1, t + 2, t + 3\}$. Posons $L(c) = [1, t + 1]$ et $R(c) = [t + 2, k]$ qui est une LR-décomposition valide. On montre qu'à la fois $L(c)$ et $R(c)$ sont cassés. Le plateau $T(c)$ peut être obtenu par des configurations $d \in \mathcal{G}_{(n)}$ telles que $c = V_i^r(d)$, pour $i \in [t, t + 3]$:
 - (i) si $i = t$ ou $i = t + 1$; alors $d_i > d_{i+1} < d_{i+2}$, impossible à cause du lemme 22 ;
 - (ii) si $i = t + 3$; alors $d_t = d_{t+1} = d - t + 2 < d_{t+3}$ impossible d'après le lemme 23 ;
 - (iii) si $i = t + 2$; alors $d_t = d_{t+1} < d_{t+2}$ et à cause du lemme 24, $L(d) = [1, t + 2]$ est cassé puisque $L(d) \cap T(d) = \emptyset$. Donc $L(c)$ est cassé car $L(c) = L(d)$ et pour tout $i \in L(c)$, $c_i = d_i$. Pour $R(c)$ il faut considérer deux sous-cas.
 - a) Soit $c_{t+3} \geq c_{t+4} + 2$; auquel cas le plateau c_{t+2}, c_{t+3} de $R(c) \cap T(c)$ est séparé de tout autre plateau potentiel de $R(c)$ par la falaise en position $t + 3$, donc $R(c)$ est cassé.
 - b) Soit $c_{t+3} = c_{t+4} + 1$; alors $d_{t+2} > d_{t+3} = c_{t+4}$ et $[t + 3, k]$ est cassé dans d (lemme 24). Cela implique que si $j \geq t + 4$ est le plus petit indice tel que $d_j = d_{j+1}$, il y a une falaise dans d à un indice h , $t + 4 \leq h < j$. Cette falaise est aussi dans c , et dans c il n'y a pas de plateau entre les positions $t + 3$ et h . Par conséquent le plateau c_{t+2}, c_{t+3} est séparé de tout autre plateau de $R(c)$ par la falaise à la position h , $R(c)$ est cassé.

On a des résultats semblables si $c = V_i^l(d)$, $i \in [t, t + 3]$. Par conséquent les deux seules valeurs possibles pour d impliquent que $L(c)$ et $R(c)$ sont cassés.

- Si $|T(c)| = 3$, soit $T(c) = \{t, t + 1, t + 2\}$. Si $X(c) = [t + 1, k]$ est cassé, soit $L(c) = [1, t]$ et $R(c) = X(c)$. C'est une LR-décomposition de c (lemme 22) qui est cassée car $L(c) \cap T(c)$ est cassé d'après le lemme 24.

Si $X(c)$ n'est pas cassé, on sait néanmoins d'après le lemme 24 que $[t + 2, k]$ l'est. Il suffit alors de prouver que $Y(c) = [1, t + 1]$ l'est aussi. Soit $j \in [t + 2, k]$ le plus petit indice tel que $c_j = c_{j+1}$. On remarque que pour tout $h \in [t + 2, j - 1]$ on a $c_h = c_{h+1} + 1$. Si $j = t + 2$, on se trouve dans le cas $|T(c)| = 4$ déjà traité. Sinon, pour tout antécédent d de c tel que $c = V_i^r(d)$, $i \in [t, j + 1]$, on a l'un des cas suivants :

- (i) si $i = t$; alors $d_t > d_{t+1} < d_{t+2}$, impossible d'après le lemme 22 ;

- (ii) si $i \in [t+1, j-1]$; alors $d_i > d_{i+1} = d_{i+2}$, impossible à cause du lemme 24 ;
- (iii) si $i = j$; alors $d_{j-1} > d_j < d_{j+1}$, impossible d'après le lemme 22 ;
- (iv) si $i = j+1$; il y a un plateau en d_{j-1}, d_j , par induction (similaire à ce qui a été fait dans la preuve du lemme 24) cela nous ramène au cas $|T(e)| = 4$ pour un ancêtre e de c , par conséquent $Y(c)$ est cassé.

Si d est tel que $c = V_i^l(d)$, $i \in \{t, t+1, t+2, j+1\}$, les possibilités sont les suivantes :

- (i) si $i = t$; alors $d_t > d_{t+1} = d_{t+2}$, impossible à cause du lemme 24 ;
- (ii) si $i = t+1$; le raisonnement est le même que pour le sous-cas (iii) du cas $|T(c)| = 4$, ce qui implique que $Y(c)$ est cassé ;
- (iii) si $i = t+2$ ou $i = j+1$; alors $d_{i-2} > d_{i-1} < d_i$ qui est impossible (lemme 22).

La LR-décomposition $(Y(c), X(c))$ est donc une LR-décomposition cassée de c . \square

On prouve la réciproque de la proposition 26 à l'aide d'un dernier lemme technique.

Lemme 27 *Pour toute configuration c telle que $c \neq (n)$ pour tout $n \in \mathbb{N}^*$, et qui admet une LR-décomposition cassée, il existe d telle que $c \in \bar{f}(d)$ qui admet une LR-décomposition cassée.*

Preuve. Posons $c = (c_1, \dots, c_k)$ avec $c \neq (n)$ où $n = \sum_{i=1}^k c_i$. Supposons que c admet une LR-décomposition cassée notée $(L(c), R(c))$. On distingue 3 cas selon le nombre d'éléments de $L(c)$.

- Si $L(c) = \emptyset$; $|R(c)| > 1$ car $c \neq (n)$. 2 cas se présentent.
 - (i) $R(c)$ ne contient pas de plateau, on construit d ainsi : $d_1 = c_1 + 1$, $d_2 = c_2 - 1$ et $d_i = c_i$ pour $i \in [3, k]$. Alors $c = V_1^r(d)$ et $(L(d) = \emptyset, R(d) = R(c))$ est une LR-décomposition cassée de d . Notons que si $c_2 = 1$, $d_2 = 0$ donc d aura pour longueur $k - 1$. Dans ce cas particulier, on définit $L(d) = \emptyset$ et $R(d) = [1, k - 1]$.
 - (ii) $R(c)$ contient au moins un plateau : soit i le plus grand indice tel que $c_i = c_{i+1}$. Soit d tel que $d_i = c_i + 1$, $d_{i+1} = c_{i+1} - 1$ et $d_j = c_j$ pour $j \in [1, k] \setminus \{i, i+1\}$. On a $c = V_i^r(d)$ et $(L(d) = L(c), R(d) = R(c))$ est une LR-décomposition cassée par hypothèse. Cette fois aussi si $c_i = 1$, d est de longueur $k - 1$, on pose alors $L(d) = \emptyset$ et $R(d) = [1, k - 1]$.
- Si $|L(c)| = 1$; on a 2 possibilités. Si $c_1 > c_2$ on revient au cas précédent, on suppose donc que $c_1 \leq c_2$. Soit d tel que $d_1 = c_1 - 1$, $d_2 = c_2 + 1$ et $d_i = c_i$ pour $i \in [3, k]$, d'où $c = V_2^l(d)$. $(L(d) = L(c), R(d) = R(c))$ est une LR-décomposition cassée de d . Là aussi si $c_1 = 1$, $d_1 = 0$ donc d doit être décalé d'un à gauche. On pose alors dans ce cas particulier $L(d) = \emptyset$ et $R(d) = [1, k - 1]$.
- Si $|L(c)| > 1$; il faut examiner 2 cas.
 - (i) $L(c)$ ne contient pas de plateau : soit une configuration d telle que $d_1 = c_1 - 1$, $d_2 = c_2 + 1$ et $d_i = c_i$ pour $i \in [3, k]$. On a $c = V_2^l(d)$ et $(L(d) = L(c), R(d) = R(c))$ est une LR-décomposition cassée de d . Si $c_1 = 1$, on a $L(d) = L(c) \setminus \{m\}$, $R(d) = (R(c) \cup \{m\}) \setminus \{k\}$ où $m = \max L(c)$.
 - (ii) $L(c)$ contient au moins un plateau : soit i le plus petit indice tel que $c_i = c_{i+1}$. Soit d tel que $d_i = c_i - 1$, $d_{i+1} = c_{i+1} + 1$ et $d_j = c_j$ pour $j \in [1, k] \setminus \{i, i+1\}$. Bien sûr, $c = V_{i+1}^l(d)$ et $(L(d) = L(c), R(d) = R(c))$ est une LR-décomposition cassée par hypothèse. Encore une fois si $i = 1$ et $c_1 = 1$, d doit être décalé à gauche. En notant $m = \max L(c)$ on a $L(d) = L(c) \setminus \{m\}$, $R(d) = (R(c) \cup \{m\}) \setminus \{k\}$. \square

La proposition suivante montre que la condition « avoir une LR-décomposition cassée » est suffisante pour appartenir à un graphe des orbites.

Proposition 28 *Si une configuration c admet une LR-décomposition cassée, alors il existe $n \in \mathbb{N}^*$ tel que $c \in \mathcal{G}_{(n)}$.*

Preuve. Soit une configuration $c = (c_1, \dots, c_k)$, avec $n = \sum_{i=1}^k c_i$, admettant une LR-décomposition cassée. Si $k = 1$, i.e. $c = (n)$ alors la preuve est terminée. Sinon, avec l'aide du lemme 27 on construit une suite de configurations $(d^0, d^1, \dots, d^h, \dots)$ telle que $d^0 = c$ et pour $h > 0$, $d^h \in \bar{f}(d^{h+1})$ et d^h admet une LR-décomposition cassée. Cette suite est finie, car pour tout h on a d'après le lemme 19 que $E(\{d^{h+1}\}) > E(f(\{d^{h+1}\})) \geq E(\{d^h\})$, et d'après le lemme 18 que $\bar{E}((n)) \geq \bar{E}(d^h)$ si $d^h \neq (n)$. Donc il existe $l \in \mathbb{N}$ tel que $d^l = (n)$, il y a donc un chemin dans $\mathcal{G}_{(n)}$ de $d^l = (n)$ à $d^0 = c$. \square

D'après la proposition 26, un point fixe π de $\mathcal{G}_{(n)}$ a une structure bien précise. Il admet une LR-décomposition cassée $(L(\pi), R(\pi))$, et puisqu'il ne contient pas de falaises à la fois $L(\pi)$ et $R(\pi)$ contiennent au plus un plateau. Cette structure est représentée sur la figure 16. Le cas où $L(\pi)$ et $R(\pi)$ se recollent à des hauteurs différentes (c'est-à-dire $|T(\pi)| = 1$) correspond à la figure 16(a), le cas où la jonction crée un plateau supplémentaire au sommet (donc $|T(\pi)| \geq 2$) est illustré sur la figure 16(b).

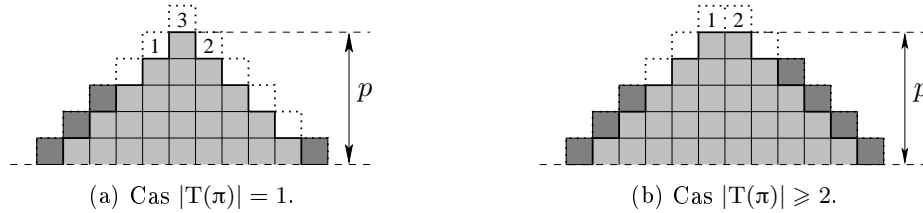


FIG. 16 – Structure des points fixes.

3.3 Points fixes

On sait depuis le corollaire 21 que pour tout $n \in \mathbb{N}^*$, la configuration (n) aboutit toujours sur un point fixe. Avec l'aide des résultats de la partie précédente nous pouvons décrire et compter les points fixes pouvant être atteints à partir de (n) .

Comme nous l'avons vu à la fin de la partie précédente (voir figures 16), intuitivement un point fixe est composé d'une pyramide de base (en gris clair sur les figures) de hauteur p à laquelle on ajoute des grains supplémentaires sur les flancs (en gris foncé) de bas en haut, à condition d'en ajouter au plus un par colonne (limite en pointillés). Les boîtes étiquetées par des numéros sont interdites, car sur la figure 16(a) remplir les cases 1 ou 2 reviendrait à considérer le cas $|T(c)| \geq 2$, et remplir la case 3 signifierait que l'on aurait dû prendre comme hauteur de pyramide la valeur $p + 1$. Dans le cas de la figure 16(b), remplir l'une ou l'autre des cases 1 et 2 nous conduit au cas $|T(c)| = 1$, remplir les deux signifie que l'on n'a pas la bonne valeur de p . À titre d'exemple, l'annexe A regroupe tous les points fixes atteignables depuis (n) pour $1 \leq n \leq 24$.

Nous ne décrivons pas plus formellement ici la structure des points fixes, lourde et sans intérêt pour cette partie, tout en notant que c'est parfaitement réalisable. Nous ne faisons que compter le nombre de points fixes correspondant aux figures 16, en commençant par montrer dans le lemme qui suit que ce sont les seuls possibles.

Pour $n \in \mathbb{N}^*$, soit $g_1(n)$ le nombre de points fixes $\pi \in \mathcal{G}_{(n)}$ tels que $|\mathbf{T}(\pi)| = 1$ et $g_2(n)$ le nombre de points fixes tels que $|\mathbf{T}(\pi)| \geq 2$.

Lemme 29 *Pour tout $n \in \mathbb{N}^*$, le nombre de points fixes de $\mathcal{G}_{(n)}$ pour SSPM est $G(n) = g_1(n) + g_2(n)$.*

Preuve. Soit $n \in \mathbb{N}^*$ et $\pi = (\pi_1, \dots, \pi_k)$ un point fixe de $\mathcal{G}_{(n)}$. Si $|\mathbf{T}(\pi)| = 1$, alors π peut être construit comme indiqué sur la figure 16(a) (au plus un plateau à gauche et un autre à droite). De plus, il ne peut être construit à partir de la figure 16(b).

Si $|\mathbf{T}(\pi)| \geq 2$, il ne peut être décrit par la figure 16(a). Par contre il peut l'être à partir de la figure 16(b), en choisissant une LR-décomposition cassée de π ($\mathbf{L}(\pi) = [1, t]$, $\mathbf{R}(\pi) = [t + 1, k]$). Si $\pi_t = \pi_{t+1}$ il suffit de découper la figure 16(b) en deux au milieu de la configuration (entre les cases 1 et 2). L'intervalle $\mathbf{L}(\pi)$ peut rentrer à gauche et $\mathbf{R}(\pi)$ à droite (au plus un plateau). Si $\pi_t = \pi_{t+1} + 1$, alors $\mathbf{T}(\pi) \subset \mathbf{L}(\pi)$. Cette fois il suffit de découper la configuration à droite de la case marquée 2, $\mathbf{L}(\pi)$ et $\mathbf{R}(\pi)$ s'adaptent parfaitement. Le cas symétrique $\pi_t = \pi_{t+1} - 1$ est semblable.

Réciproquement, toutes les configurations à n grains qui peuvent être construites d'après les figures 16 sont des points fixes qui admettent une LR-décomposition cassée, donc d'après la proposition 28 sont des points fixes de $\mathcal{G}_{(n)}$. L'ensemble des points fixes de (n) est donc exactement représenté sur les figures 16, et $G(n) = g_1(n) + g_2(n)$. \square

Les deux lemmes qui suivent donnent l'expression exacte des fonctions $g_1(n)$ et $g_2(n)$.

Lemme 30 *Pour tout $n \in \mathbb{N}^*$, le nombre de points fixes π de $\mathcal{G}_{(n)}$ tels que $|\mathbf{T}(\pi)| = 1$ est*

$$g_1(n) = \begin{cases} n - p^2 + 1 & \text{si } n - p^2 \leq p - 1, \\ 2p - n + p^2 - 1 & \text{si } p \leq n - p^2 \leq 2p - 1, \\ 0 & \text{sinon,} \end{cases}$$

où p est l'unique entier tel que $p^2 \leq n < (p + 1)^2$.

Preuve. Soit $n \in \mathbb{N}^*$ et un point fixe π de $\mathcal{G}_{(n)}$. Puisque par hypothèse $|\mathbf{T}(\pi)| = 1$, π a la structure illustrée sur la figure 16(a). Comme n est connu, on peut déterminer p . C'est le seul entier tel que $p^2 \leq n < (p + 1)^2$ (p^2 est la surface de la pyramide gris clair), soit $p = \lfloor \sqrt{n} \rfloor$. Notons $u = n - p^2 < 2p + 1$ le nombre de grains excédentaires à placer. L'ensemble des manières de placer ces u grains donne tous les points fixes possibles (lemme 29), il y a 3 façons de le faire.

(i) Si $0 \leq u < p$; tous les grains sont placés à gauche, de bas en haut, on obtient un point fixe. Ensuite on ne place plus que $u - 1$ grains à gauche et un à droite, pour en obtenir un autre. On itère ce procédé jusqu'à mettre les u grains à droite, cela donne $u + 1$ points fixes.

(ii) Si $p \leq u \leq 2p - 2$; on commence avec $p - 1$ grains à gauche et les $u - p + 1$ qui restent à droite, toujours de bas en haut, pour obtenir le premier point fixe. Un autre est obtenu en mettant $p - 2$ grains à gauche et $u - p + 2$ à droite, et ainsi de suite jusqu'à en avoir $p - 1$ à droite. Cette procédure donne lieu à $(p - 1) - (u - p + 1) + 1 = 2p - u - 1$ points fixes distincts.

- (iii) Si $u = 2p - 1$ ou $u = 2p$; il y a forcément p grains à mettre à droite ou à gauche, ce qui n'est pas possible (les cases 1 et 2 de la figure 16(a) sont interdites), il n'y a alors aucun point fixe. \square

Lemme 31 Pour tout $n \in \mathbb{N}^*$, le nombre de points fixes π de $\mathcal{G}_{(n)}$ tels que $|\mathbf{T}(\pi)| \geq 2$ est

$$g_2(n) = \begin{cases} n - p^2 - p + 1 & \text{si } n - p^2 - p \leq p - 1, \\ p & \text{si } n - p^2 - p \leq p, \\ 3p - n + p^2 + 1 & \text{si } p + 1 \leq n - p^2 - p \leq 2p + 1, \end{cases}$$

où p est l'unique entier tel que $p^2 + p \leq n < (p + 1)^2 + (p + 1)$.

Preuve. Cette preuve est semblable à celle du lemme 30, à part qu'elle s'appuie sur la figure 16(b) qui représente les points fixes de sommet plus grand que 2. Soit $n \in \mathbb{N}^*$ et π un point fixe de $\mathcal{G}_{(n)}$ tel que $|\mathbf{T}(\pi)| \geq 2$. La hauteur de π est l'unique entier p tel que $p^2 + p \leq n \leq (p + 1)^2 + (p + 1)$ (surface de la pyramide grise), soit $p = \lfloor \frac{1}{2}(\sqrt{4n + 1} - 1) \rfloor$. Soit $v = n - p^2 - p < 2p + 2$ le nombre de grains restant à répartir pour obtenir tous les points fixes (lemme 29), selon la valeur de v il y a encore une fois 3 cas.

- (i) Si $0 \leq v < p$; si l'on met les v grains à gauche et aucun à droite on obtient le premier point fixe. On n'en met ensuite plus que $v - 1$ à gauche et un à droite, et ainsi de suite pour obtenir $v + 1$ points fixes.
- (ii) Si $v = p$; on peut mettre p grains à gauche et 0 à droite, puis $p - 1$ à gauche et 1 à droite, et ainsi de suite jusqu'à 0 à gauche et p à droite. Il devrait donc y avoir $p + 1$ points fixes, mais remarquons que le point fixe avec p grains sur la gauche et celui avec p grains sur la droite sont identiques : ils forment un point fixe de sommet de longueur 3, sans plateaux. Il n'y a donc que p points fixes *différents*, c'est le seul cas de doublon qui se présente.
- (iii) Si $p < v \leq 2p + 1$; on met p grains à gauche et les $v - p$ restants à droite, puis $p - 1$ à gauche et $v - p + 1$ à droite, jusqu'à $v - p$ à gauche et p à droite. Cela donne $p - (v - p) + 1 = 2p - v + 1$ points fixes dans ce cas. \square

La proposition suivante donne une formule close pour exprimer le nombre de points fixes de $\mathcal{G}_{(n)}$ à l'aide des lemmes précédents. La formule est un peu « magique », le résultat est étonnamment simple par rapport aux calculs effectués qui se simplifient entièrement. De plus nous n'avons pas d'interprétation visuelle ou intuitive pour un tel résultat.

Proposition 32 Pour tout $n \in \mathbb{N}^*$, le nombre de points fixes de $\mathcal{G}_{(n)}$ pour SSPM est $G(n) = \lfloor \sqrt{n} \rfloor$.

Preuve. Soient $n \in \mathbb{N}^*$ et $p = \lfloor \sqrt{n} \rfloor$ l'unique entier vérifiant $p^2 \leq n < (p + 1)^2$. On distingue 3 cas qui correspondent aux cas des lemmes 30 et 31.

- (i) Si $p^2 \leq n \leq p^2 + p - 1$; alors $(p - 1)^2 + (p - 1) \leq n < p^2 + p$. On se trouve dans le cas (i) du lemme 30 et dans le cas (iii) du lemme 31, donc d'après le lemme 29 on a $G(n) = [n - p^2 + 1] + [3(p - 1) - n + (p - 1)^2 + 1] = p = \lfloor \sqrt{n} \rfloor$.
- (ii) Si $p^2 + p \leq n \leq p^2 + 2p - 1$; on est dans le cas (ii) du lemme 30 et dans le cas (i) du lemme 31. D'après le lemme 29 on a $G(n) = [2p - n + p^2 - 1] + [n - p^2 - p + 1] = p = \lfloor \sqrt{n} \rfloor$.
- (iii) Si $n = p^2 + 2p$; c'est le cas (iii) du lemme 30 et le (ii) du lemme 31, d'où $G(n) = 0 + p = \lfloor \sqrt{n} \rfloor$. \square

3.4 Longueur du transitoire

Nous venons de voir que SSPM avait une dynamique de type point fixe, et que seuls $\lfloor \sqrt{n} \rfloor$ points fixes (de forme assez proche) étaient atteignables. Il est maintenant naturel de se demander combien d'itérations sont nécessaires pour les atteindre ; cela peut être vu comme une façon d'estimer le « temps » que dure une avalanche de grains.

De manière générale la *longueur du transitoire* (relativement à une configuration initiale c et à un point fixe $\pi \in \mathcal{G}_c$) est la longueur d'une orbite de c à π dans \mathcal{G}_c . Comme il y a plusieurs points fixes et plusieurs chemins pour les atteindre, toutes ces longueurs ne seront pas toujours identiques. On essaie donc dans cette partie d'estimer les longueurs *minimales* et *maximales* des transitoires, *i.e.* les longueurs des orbites respectivement les plus courtes et les plus longues de c à π dans \mathcal{G}_c .

Nous prouvons dans un premier temps que la longueur minimale du transitoire de (n) est en $\mathcal{O}(n^{3/2})$ comme pour SPM (voir proposition 6 et remarque 2 page 12), en exhibant un des chemins de longueur minimale. Puis nous montrons dans une seconde partie que la longueur maximale du transitoire de (n) est au plus en $\mathcal{O}(n^2)$, et vraisemblablement en $\mathcal{O}(n^{3/2})$ également. Ce travail a été fait pour l'essentiel dans [25].

3.4.1 Longueur minimale du transitoire

On s'intéresse donc de nouveau aux configurations initiales à une colonne, de type (n) . Pour tout $n \in \mathbb{N}^*$ et tout point fixe $\pi \in \mathcal{G}_{(n)}$ on note $t(\pi)$ la longueur minimale du transitoire de (n) à π , c'est-à-dire que $t(\pi)$ est la longueur du chemin le plus court reliant (n) à π dans $\mathcal{G}_{(n)}$.

Nous avons besoin dans cette partie de retenir la position de la colonne initiale, pour pouvoir savoir où elle est relativement au point fixe π qui nous intéresse. Contrairement à la partie précédente (voir remarque 6) les indices des colonnes ne sont pas décalés lors d'une application d'une règle à une extrémité de la configuration.

En pratique, si l'on note $\pi = (\pi_1, \dots, \pi_k)$ un point fixe de (n) , toutes les configurations entre (n) et π peuvent être définies sur le même ensemble d'indices $[1, k]$ car les règles locales ne peuvent faire rétrécir une configuration. La figure 17 illustre ceci, les colonnes sont numérotées à partir de leur position dans le point fixe final.

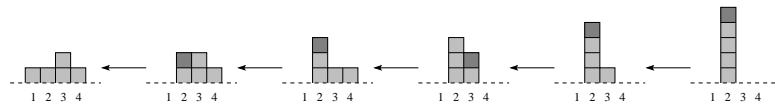


FIG. 17 – Une orbite de (5) pour SSPM, où les positions des colonnes par rapport au point fixe visé sont explicitées. Les grains qui tombent sont en gris foncé, on remarque que la colonne initiale est à l'indice 2 relativement au point fixe.

On définit le *centre* d'une configuration c relativement à un point fixe $\pi = (\pi_1, \dots, \pi_k) \in \mathcal{G}_c$ comme suit :

$$\gamma_\pi(c) = \min \left\{ i \in [1, k], \sum_{j=1}^i c_j \geq \sum_{j=i+1}^k c_j \right\} .$$

Intuitivement, le centre d'une configuration permet de « couper » en deux une configuration, de manière à ce qu'il y ait presque le même nombre de grains dans la partie de gauche et dans la partie de droite.

Notons qu'il se peut que pour une configuration c donnée et un de ses points fixes $\pi \in \mathcal{G}_c$ on ait $\gamma_\pi(c) \neq \gamma_\pi(\pi)$. Par exemple sur la figure 17, $\gamma_\pi((5)) = 2 \neq \gamma_\pi(\pi) = 3$.

Pour tout point fixe $\pi \in \mathcal{G}_{(n)}$, soit $t_i(\pi)$ la longueur minimale d'un chemin de (n) à π tel que $\gamma_\pi((n)) = i$. La longueur minimale du transitoire est donc $t(\pi) = \min_{i \in [1, k]} t_i(\pi)$ (forcément $\gamma_\pi((n)) \in [1, k]$). Le résultat suivant relie la longueur minimale du transitoire à la position de la colonne initiale $\gamma_\pi((n))$.

Lemme 33 *Pour tout $n \in \mathbb{N}^*$, pour tout point fixe $\pi \in \mathcal{G}_{(n)}$, $t(\pi) = t_{\gamma_\pi(\pi)}(\pi)$ i.e. la longueur du transitoire est minimale quand la colonne initiale était placée en $\gamma_\pi(\pi)$.*

Preuve. Soit $n \in \mathbb{N}^*$. Soit $\pi = (\pi_1, \dots, \pi_k) \in \mathcal{G}_{(n)}$ un point fixe et $i = \gamma_\pi((n))$. Un grain de la colonne j a besoin d'au moins $|i - j|$ itérations pour atteindre la colonne j depuis la colonne i . En sommant sur tous les grains,

$$t_i(\pi) = \sum_{j=1}^k |i - j| \pi_j$$

est la longueur minimale du transitoire si $\gamma_\pi((n)) = i$, donc

$$t_{i+1}(\pi) - t_i(\pi) = \sum_{j=1}^k (|i - j + 1| - |i - j|) \pi_j = \sum_{j=1}^i \pi_j - \sum_{j=i+1}^k \pi_j .$$

Par conséquent, par définition de $\gamma_\pi(\pi)$, la fonction $i \rightarrow t_i(\pi)$ est strictement décroissante pour $i \leq \gamma_\pi(\pi)$ et croissante pour $i \geq \gamma_\pi(\pi)$. Le minimum est donc bien atteint pour $i = \gamma_\pi(\pi)$. \square

Le lemme suivant permet de comprendre pourquoi γ_π s'appelle le « centre ». En effet pour un point fixe π , on montre que $\gamma_\pi(\pi)$ correspond à une des colonnes les plus centrales du sommet de π .

Lemme 34 *Pour tout $n \in \mathbb{N}^*$, pour tout point fixe $\pi \in \mathcal{G}_{(n)}$,*

$$\gamma_\pi(\pi) = \begin{cases} u & \text{si } |T(\pi)| = 1 \\ u + 1 & \text{ou si } |T(\pi)| = 2 \text{ et } \sum_{j=1}^u \pi_j \geq \sum_{j=u+1}^k \pi_k, \\ & \text{sinon,} \end{cases}$$

avec $u = \min T(\pi)$.

Preuve. Soit $n \in \mathbb{N}^*$, soit un point fixe $\pi = (\pi_1, \dots, \pi_k) \in \mathcal{G}_{(n)}$ de hauteur maximale p et soit $u = \min T(\pi)$. On montre dans un premier temps que $\gamma_\pi(\pi) \geq u$, en effet si $\gamma_\pi(\pi) = u - 1$ alors

$$\sum_{j=1}^{\gamma_\pi(\pi)} \pi_j = \sum_{j=1}^{p-1} j + x < \sum_{j=1}^p j \leq \sum_{j=\gamma_\pi(\pi)+1}^k \pi_j ,$$

où $x \in [0, p - 1]$ est la hauteur du plateau de gauche de π , comme indiqué sur la figure 18. On a également $\gamma_\pi(\pi) \notin [1, u - 2]$ car la fonction $i \rightarrow \left(\sum_{j=1}^i c_j - \sum_{j=i+1}^k c_j \right)$ est croissante pour tout c de longueur k .

Nous distinguons maintenant quatre cas, selon la taille de $T(\pi)$. Pour les deux premiers, on note $x \in [0, p - 1]$ la hauteur du plateau de $L(\pi)$ et $y \in [0, p - 1]$ la hauteur du plateau de $R(\pi)$. Si l'un de ces plateaux n'existe pas il suffit de poser $x = 0$ ou $y = 0$.

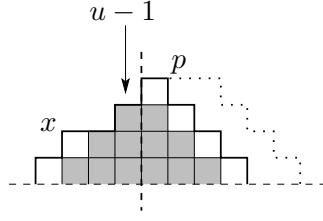


FIG. 18 – Situation lorsque $\gamma_\pi(\pi) = u - 1$. Il y a moins de grains à gauche qu'à droite.

- Si $|\mathbf{T}(\pi)| = 1$; alors $\gamma_\pi(\pi) = u$ puisque

$$\sum_{j=1}^u \pi_j = \sum_{j=1}^p j + x > \sum_{j=1}^{p-1} j + y = \sum_{j=u+1}^k \pi_j .$$

- Si $|\mathbf{T}(\pi)| = 2$; si $\sum_{j=1}^u \pi_j \geq \sum_{j=u+1}^k \pi_j$ alors clairement $\gamma_\pi(\pi) = u$. Sinon, $\sum_{j=1}^u \pi_j < \sum_{j=u+1}^k \pi_j$ implique que $x < y$ et

$$\sum_{j=1}^{u+1} \pi_j - \sum_{j=u+2}^k \pi_j = 2p + x - y > 0 ,$$

d'où $\gamma_\pi(\pi) = u + 1$.

- Si $|\mathbf{T}(\pi)| = 3$; alors

$$\sum_{j=1}^u \pi_j - \sum_{j=u+1}^k \pi_j = -p \pm x < 0 \quad \text{et} \quad \sum_{j=1}^{u+1} \pi_j - \sum_{j=u+2}^k \pi_j = p \pm x > 0 ,$$

où $x \in [0, p - 1]$ est la hauteur de l'unique plateau de π (soit dans $\mathbf{L}(\pi)$ soit dans $\mathbf{R}(\pi)$), donc $\gamma_\pi(\pi) = u + 1$.

- Si $|\mathbf{T}(\pi)| = 4$; alors

$$\sum_{j=1}^u \pi_j - \sum_{j=u+1}^k \pi_j = -2p < 0 \quad \text{et} \quad \sum_{j=1}^{u+1} \pi_j - \sum_{j=u+2}^k \pi_j = 0 ,$$

d'où $\gamma_\pi(\pi) = u + 1$. □

On peut déduire de ces deux lemmes la valeur exacte de la longueur minimale du transitoire.

Proposition 35 *Pour tout $n \in \mathbb{N}^*$, pour tout point fixe $\pi \in \mathcal{G}_{(n)}$, la longueur minimale du transitoire de (n) à π est*

$$t(\pi) = \begin{cases} \frac{1}{3}(p-1)p(p+1) + \frac{1}{6}x(x+1)(3p-2x+2) + \frac{1}{6}y(y+1)(3p-2y+2) \\ \quad \text{si } |\mathbf{T}(\pi)| = 1, \text{ avec } p = \lfloor \sqrt{n} \rfloor \text{ et } 0 \leq x, y \leq p-1, \\ \frac{1}{3}(p-1)p(p+1) + \frac{1}{6}x(x+1)(3p-2x+2) + \frac{1}{6}y(y+1)(3p-2y+2) \\ \quad + \frac{1}{2}p(p+1) + \frac{1}{2}x(x+1) \\ \quad \text{sinon, avec } p = \lfloor \frac{1}{2}(\sqrt{4n+1}-1) \rfloor \text{ et } 0 \leq x \leq y \leq p, \end{cases}$$

où p est la hauteur de π et x, y les hauteurs respectives des deux (éventuels) plateaux de $\mathbf{L}(\pi)$ et de $\mathbf{R}(\pi)$.

Preuve. Soit $n \in \mathbb{N}^*$, le point fixe $\pi = (\pi_1, \dots, \pi_k) \in \mathcal{G}_{(n)}$ a la structure de l'une des figures 16. Le lemme 33 nous dit que $t_i(\pi)$ est minimal lorsque $i = \gamma_\pi(\pi)$. Il faut distinguer 2 cas, si $|\mathbf{T}(\pi)| = 1$ (figure 16(a)) alors d'après le lemme 34 on a $\gamma_\pi(\pi) = \min \mathbf{T}(\pi)$. Par conséquent,

$$\begin{aligned} t(\pi) &= t_{\gamma_\pi(\pi)}(\pi) = \sum_{j=1}^k |\gamma_\pi - j| \pi_j \\ &= 2 \sum_{j=1}^{p-1} j(p-j) + \sum_{j=1}^x j(p-j+1) + \sum_{j=1}^y j(p-j+1) , \end{aligned}$$

où $x, y \in [0, p-1]$ sont les hauteurs des plateaux de $\mathbf{L}(\pi)$ et $\mathbf{R}(\pi)$; p est la hauteur de π , c'est l'unique entier qui satisfait $p^2 \leq n < (p+1)^2$, soit $p = \lfloor \sqrt{n} \rfloor$. Le résultat correspond au développement de cette équation.

Dans le cas où $|\mathbf{T}(\pi)| \geq 2$ (figure 16(b)), on note $x, y \in [0, p]$ les hauteurs des plateaux de $\mathbf{L}(\pi)$ et $\mathbf{R}(\pi)$, avec $x \leq y$. On suppose que y est la hauteur du plateau de $\mathbf{L}(\pi)$ (sinon la longueur du transitoire est identique pour des raisons de symétrie). D'après le lemme 34, $\gamma_\pi(\pi)$ est la colonne centrale de $\mathbf{T}(\pi)$ si $|\mathbf{T}(\pi)|$ est impair, ou la colonne de gauche des deux colonnes centrales de $\mathbf{T}(\pi)$ si $|\mathbf{T}(\pi)|$ est pair (cela correspond dans tous ces cas à la colonne sous la case marquée d'un 1 sur la figure 16(b)). Alors on a

$$\begin{aligned} t(\pi) &= t_{\gamma_\pi(\pi)}(\pi) = \sum_{j=1}^k |\gamma_\pi - j| \pi_j \\ &= 2 \sum_{j=1}^{p-1} j(p-j) + \sum_{j=1}^x j(p-j+1) + \sum_{j=1}^y j(p-j+1) + \sum_{j=1}^p j + \sum_{j=1}^x j , \end{aligned}$$

où p est la hauteur de π , c'est l'unique entier vérifiant $p(p+1) \leq n < (p+1)(p+2)$, soit $p = \lfloor \frac{1}{2}(\sqrt{4n+1} - 1) \rfloor$. On obtient le résultat attendu en développant cette équation.

Il est évident que ce nombre minimal d'itérations est nécessaire. Il existe une orbite qui atteint π à partir de (n) en $t(\pi)$ étapes : supposons que $\gamma_\pi(\pi) \in \mathbf{L}(\pi)$, on commence par faire s'écrouler les grains de la colonne initiale sur la droite (règle SPM), jusqu'à ce que $\mathbf{R}(\pi)$ soit formée. L'union de la colonne initiale et de $\mathbf{R}(\pi)$ est cassée, donc est constructible par SPM (lemme 2 page 11). Ensuite on fait tomber les grains restants sur la gauche avec la règle symétrique de celle de SPM, jusqu'à obtenir π (c'est possible car $\mathbf{L}(\pi)$ est cassé). Si $\gamma_\pi(\pi) \in \mathbf{R}(\pi)$ on effectue la même construction en commençant par $\mathbf{L}(\pi)$. Chaque grain de chaque colonne j a eu besoin de $|\gamma_\pi(\pi) - j|$ itérations, en sommant sur j on obtient bien $t(\pi)$. \square

Remarque 7 Comme on a dans tous les cas $p = \mathcal{O}(n^{1/2})$ et comme $x, y \leq p$, au moins $t(\pi) = \mathcal{O}(n^{3/2})$ itérations sont nécessaires pour atteindre un point fixe π à n grains à partir de (n) . On peut remarquer que c'est le même ordre de grandeur que pour SPM (proposition 6 et remarque 2, page 12).

3.4.2 À propos de la longueur maximale du transitoire

Pour tout $n \in \mathbb{N}^*$ et tout point fixe $\pi \in \mathcal{G}_{(n)}$ on note $\tau(\pi)$ la longueur de la plus longue orbite reliant (n) à π dans $\mathcal{G}_{(n)}$. Il est plus difficile de calculer exactement la longueur maximale du transitoire $\tau(\pi)$, la proposition suivante en donne une borne triviale.

Proposition 36 *Pour tout $n \in \mathbb{N}^*$ et tout point fixe $\pi \in \mathcal{G}_{(n)}$, la longueur maximale du transitoire $\tau(\pi)$ est bornée par $\mathcal{O}(n^2)$.*

Preuve. Soit $n \in \mathbb{N}^*$ et π un point fixe de $\mathcal{G}_{(n)}$. Lorsqu'un grain tombe, sa hauteur est réduite d'au moins 1. Un grain à hauteur initiale i peut donc se déplacer au plus i fois, donc $\tau(\pi) \leq \sum_{i=0}^{n-1} i = \frac{1}{2}n(n-1)$. \square

Cette borne est très grossière, non seulement parce que les grains tombent en général d'une hauteur supérieure à 1, mais également parce qu'ils ne tombent pas tous jusqu'en bas.

Intuitivement, le comportement qui demande le plus de temps serait le suivant : les grains tombent d'un côté autant que possible, puis ils s'effondrent de l'autre pour donner un point fixe. Cette orbite est encore en $\mathcal{O}(n^{3/2})$, nous conjecturons qu'elle est de longueur maximale.

Conjecture 37 *Pour tout $n \in \mathbb{N}^*$ et tout point fixe $\pi \in \mathcal{G}_{(n)}$, la longueur maximale du transitoire est $\tau(\pi) = \mathcal{O}(n^{3/2})$.*

Expérimentalement, il semble que $t(\pi) \leq \tau(\pi) \leq 2t(\pi)$, et même que $\tau(\pi) \approx 1.5t(\pi)$ pour de faibles valeurs de n (voir figure 19). Pour $n > 60$, la simulation complète de SSPM est trop longue à réaliser sur les ordinateurs dont nous disposons mais il est probable que l'on reste dans le même ordre de grandeur.

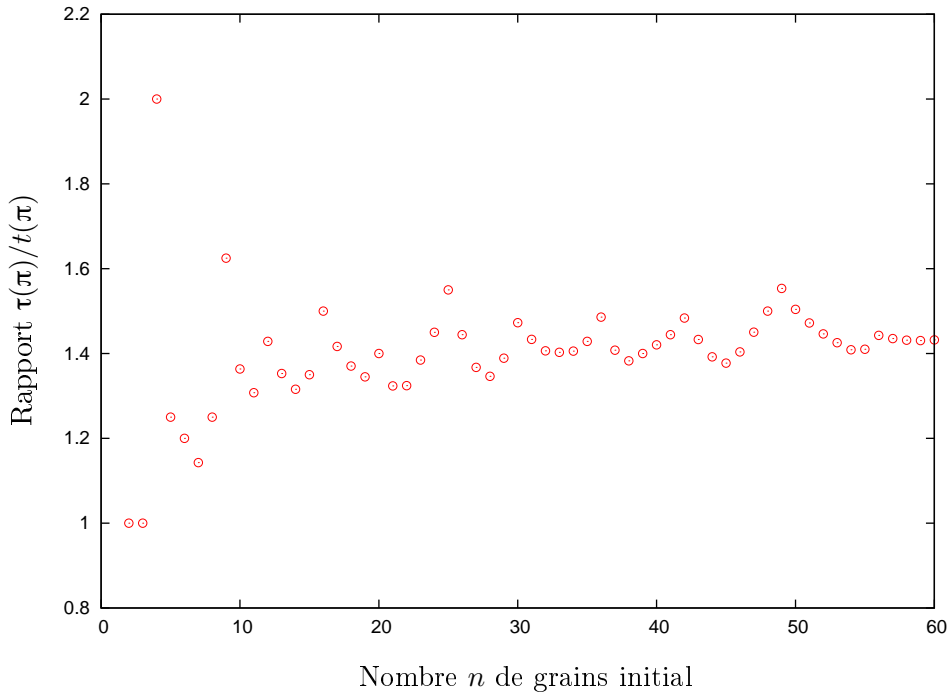


FIG. 19 – Variation du rapport $\frac{\tau(\pi)}{t(\pi)}$ en fonction de n pour SSPM.

3.5 Perspectives

Nous avons montré dans les trois parties précédentes que le modèle SSPM était un modèle intéressant pour simuler les piles de sable multi-directionnelles, et par extension multi-

dimensionnelles (ajouter une dimension revient à ajouter plusieurs directions). Bien que non-déterministe et conduisant à plusieurs états stables différents, il est possible de caractériser précisément son comportement : états intermédiaires, points fixes, longueur du transitoire sont parfaitement connus dans le cas d'une pile initiale à une colonne, en temps séquentiel. Il semble également être assez proche de la réalité, les points fixes étant des piles pyramidales réalistes et proches les unes des autres.

Ce modèle semble donc être un bon point de départ à l'étude des piles de sable dans un cadre un peu plus général que SPM ou $\text{IPM}(k)$. SSPM peut naturellement s'étendre à des modèles $\text{SIPM}(k)$. Son étude dans les mêmes conditions que ce que nous avons fait pour SSPM donnerait des résultats similaires, à savoir une caractérisation relativement simple des configurations atteignables, et un nombre de points fixes dépendant de manière simple de n et de k (expérimentalement on obtient que le nombre de points fixes $G(n)$ est le plus grand entier vérifiant $k \cdot G(n)^2 - (k - 1) \cdot G(n) \leq n$). Ceci pourrait alors ouvrir la voie à des modèles encore plus généraux, afin de pouvoir enfin simuler des tas de sable en 3 dimensions.

Cette étude mériterait encore d'être prolongée, dans un premier temps en regardant ce qu'il se passe avec une *configuration initiale quelconque* comme nous l'avons fait pour SPM et $\text{IPM}(k)$ dans le chapitre 2. Utiliser un algorithme proche de celui que l'on y a introduit ne marcherait pas : on le voit sur la figure 20, si l'on part de deux colonnes isolées, l'algorithme découpe quelque part entre les deux. Les deux intervalles s'effondrent en parallèle (première phase de calcul), et dans tous les cas l'effondrement de la première colonne reste coincé contre celui de la seconde, on obtient deux sommets distincts (figure du haut). Le comportement suivant serait perdu : la colonne de droite s'écroule entièrement, puis la première s'écroule sur la droite et recouvre la seconde, on obtient un sommet unique (en bas).

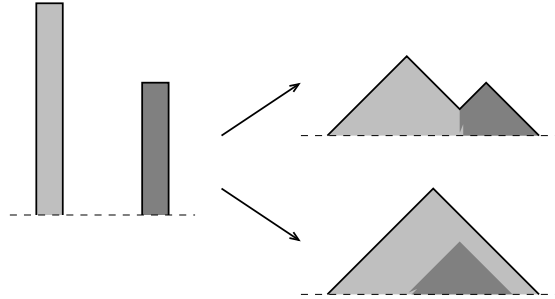


FIG. 20 – Piles de sables symétriques généralisées.

Il faudrait donc entièrement reprendre le problème pour trouver un moyen d'accélérer la simulation sans perdre aucune possibilité.

Une autre piste de recherche intéressante est l'étude de variantes de SSPM qui seraient *parallèles* (à chaque instant, tous les grains pouvant tomber tombent simultanément) ou *déterministes* (une seule orbite, un seul point fixe). Le modèle parallèle est simple à définir, et reste non-déterministe à cause des grains pouvant tomber des deux côtés à la fois. Il serait intéressant de voir si beaucoup de comportements disparaissent, et de combien la simulation est accélérée car ce mode serait encore plus réaliste.

D'autre part, un exemple de modèle déterministe est le modèle dans lequel un grain tombe de chaque côté lorsque les deux différences à gauche et à droite sont supérieures à 2. Ce modèle a un comportement semblable à SPM en mode parallèle et n'est probablement pas très intéressant.

Un autre système de règles déterministe inspiré par SSPM est celui où un grain tombe du côté où la falaise est la plus grande, et en cas d'égalité à droite. Il faudrait regarder quel point fixe (unique désormais) est obtenu parmi les $\lfloor \sqrt{n} \rfloor$ possibles.

Beaucoup de variantes sont donc envisageables, dans chaque cas il faudrait s'intéresser au graphe des orbites obtenu afin de caractériser le(s) point(s) fixe(s).

Deuxième partie

Automates de sable

1

Définitions

Sommaire

1.1	Une topologie pour les configurations	48
1.2	Automates de sable	52
1.3	Relation avec les automates cellulaires	53
1.4	Caractérisation « à la Hedlund »	56

Dans cette seconde grande partie, nous étudions les *automates de sable* introduits récemment dans [8] et dont nous avons poursuivi l'étude dans [9, 10]. Il s'agit d'un système dynamique discret au fonctionnement proche de celui des automates cellulaires [37]. Le but des automates cellulaires est de modéliser simplement un phénomène mal connu afin de mieux comprendre son fonctionnement. Pour cela, dans un premier temps on discrétise l'espace étudié selon une grille \mathbb{Z}^d , chaque point de la grille contenant un état dont la valeur est bornée. Puis chaque point de la grille évolue (simultanément) en fonction des valeurs contenues dans les états de son voisinage. Les automates cellulaires ont un fonctionnement *local* : on modélise des phénomènes qui n'ont pas une vue d'ensemble de la situation, mais qui ont plutôt besoin de bien connaître leur environnement proche. Ce principe très simple leur permet de modéliser une grande variété de phénomènes, physiques, économiques, sociaux, *etc.*, ce qui leur vaut d'avoir été énormément étudiés car ils engendrent des comportements complexes, qui diffèrent selon l'automate choisi (voir par exemple [37] pour une présentation plus détaillée des automates cellulaires).

Les automates de sable sont une version un peu modifiée des automates cellulaires. La localité en reste le principe de base mais ils agissent sur un ensemble d'états infinis. Pour que malgré cela la règle locale puisse être décrite par une table finie, un voisinage n'est pas un ensemble d'états mais un ensemble de différences avec l'état du point à modifier. Ce système dynamique est donc parfaitement adapté à la simulation des piles de sable (voir partie I) : les configurations de l'automate sont les configurations de la pile, c'est-à-dire des entiers répartis sur une grille \mathbb{Z}^d (on permet ainsi des configurations de taille infinie). La table de transition de l'automate contient des règles qui provoquent la chute de grains lorsque la différence de hauteur avec ses voisins est suffisante.

L'intérêt de l'étude de ce système dans le cadre des piles de sable est double. D'une part, il permet de simuler et d'*unifier* tous les modèles de piles de sable imaginables, pourvu qu'ils fonctionnent de manière locale. Il suffit de changer la table de transition pour simuler les règles locales du modèle visé, la seule contrainte étant un fonctionnement déterministe (mode parallèle pour SPM et IPM(k), modèle « adapté » pour SSPM) plus réaliste. D'autre part, il repose sur des

bases mathématiques très solides. Une *topologie* peut être mise sur l'espace des configurations, cet espace est alors localement compact et un automate de sable est une fonction continue de cet ensemble dans lui-même. Ces résultats permettent une étude de la dynamique des automates de sable approfondie, menant à des résultats généraux. On évite ainsi les preuves combinatoires et algébriques, assez laborieuses, telles qu'on les a faites dans la partie I pour des résultats plus ciblés.

Dans ce chapitre nous donnons les notions permettant de définir et de comprendre le fonctionnement des automates de sable. Nous rappelons les résultats topologiques de [9], en définissant une distance sur les configurations. Puis nous définissons les automates de sable, qui sont en quelque sorte un système qui agit sur les configurations en préservant la propriété qu'elles n'ont pas de « trous » (*i.e.* entre deux grains de la même colonne, il n'y a pas de vide). Nous verrons que ce système est équivalent (au sens de la simulation) aux automates cellulaires et donc aux machines de Turing. Enfin, la dernière partie présente un résultat majeur (théorème 47) pour caractériser les automates de sable. De manière similaire à ce qu'a fait Hedlund [32] pour les automates cellulaires, on montre que les automates de sable sont exactement les fonctions continues de l'espace des configurations dans lui-même qui commutent avec les deux décalages (horizontal et vertical) et qui conservent les états infinis.

1.1 Une topologie pour les configurations

Dans cette partie nous introduisons une distance entre configurations, qui induit une topologie sur l'espace des configurations. La version originale de ces définitions se trouve dans [8], celles que l'on utilise sont étendues à des dimensions d quelconques et viennent de [9] et [10].

Une *configuration* représente un ensemble de grains organisés en colonnes, elles-mêmes réparties sur la grille \mathbb{Z}^d . À chaque point de la grille est associé un *nombre de grains*, c'est-à-dire un élément de $\tilde{\mathbb{Z}} = \mathbb{Z} \cup \{+\infty, -\infty\}$. Les valeurs $+\infty$ et $-\infty$ représentent respectivement une *source* et un *puits* de grains. Formellement, une configuration x est donc un élément de $\tilde{\mathbb{Z}}^{\mathbb{Z}^d}$. L'ensemble des configurations est noté $\mathfrak{C} = \tilde{\mathbb{Z}}^{\mathbb{Z}^d}$.

Pour toute configuration $x \in \mathfrak{C}$, on note x_i ou x_{i_1, \dots, i_d} le nombre de grains dans la *colonne* de x indexée par le vecteur $i = (i_1, \dots, i_d)$. Lorsqu'il n'y a pas d'ambiguïté, un simple 0 remplace le vecteur nul $(0, \dots, 0)$. Ainsi pour tout $u \in \tilde{\mathbb{Z}}$, l'ensemble $\{x \mid x_0 = u\}$ des configurations dont l'élément central vaut u est noté \mathfrak{C}_u . Nous verrons que ces ensembles jouent un rôle très important dans l'espace des configurations, car ils sont compacts (proposition 40).

Afin de « comparer » deux vecteurs $i, j \in \mathbb{Z}^d$, on note $i \preceq j$ le fait que pour tout $k \in [1, d]$, $i_k \leq j_k$. Si $i \preceq j$ et $i \neq j$ alors on utilise la notation $i \prec j$. Enfin, pour tout vecteur i de dimension d , on note $|i| = \max_{j=1, \dots, d} |i_j|$ la norme infinie de i .

Commençons par expliquer comment fonctionne la distance que nous choisissons. Deux configurations sont d'autant plus proches qu'elles ont plus de valeurs communes au centre de la grille. Par exemple deux configurations x et y telles que $x_0 \neq y_0$ sont à distance 1. Si leur valeur centrale est identique, on place deux « observateurs » en haut des colonnes x_0 et y_0 pour mesurer les différences $x_i - x_0$ et $y_i - y_0$. Ces observateurs ont une vision limitée par un entier r appelé la *précision*, à la fois vers les côtés (ils ne peuvent voir que les voisins x_i et y_i tels que $|i| \leq r$) et vers le haut et le bas (si $|x_i - x_0| > r$ ou $|y_i - y_0| > r$ alors ils ne peuvent mesurer précisément et renvoient la valeur $+\infty$ ou $-\infty$). La figure 21 montre un exemple de mesure effectuée par un observateur. Faire ce type de mesure revient à se placer au sommet d'une montagne un jour de

brouillard : on ne voit qu'à une certaine distance et les distances verticales sont bornées par la mer de nuages en bas et les nuages en haut.

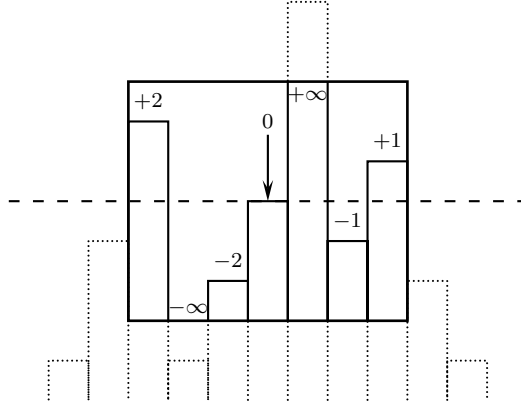


FIG. 21 – Exemple de mesure effectuée par un observateur placé en haut de la colonne d'indice 0, avec une précision de 3 cases, pour une configuration en dimension 1.

La distance entre x et y sera alors 2^{-r} , où r est le plus petit entier tel que les observateurs voient une différence entre x et y du haut de leur point de repère avec une précision de r . Cette définition permet d'obtenir des propriétés fortes, et en même temps elle est bien adaptée à la simulation des piles de sable, basées sur les différences entre colonnes voisines.

Afin de formaliser ces notions, on s'appuie sur les notations suivantes. À partir de la définition usuelle d'un intervalle d'entiers $[a, b] = \{a, a + 1, \dots, b\}$, on définit l'ensemble $\widetilde{[a, b]} = [a, b] \cup \{+\infty, -\infty\}$ qui y ajoute les infinis.

Définition 1 Un observateur (également appelé outil de mesure) de précision $r \in \mathbb{N}^*$, de hauteur de référence $m \in \widetilde{\mathbb{Z}}$, est la fonction $\beta_r^m : \widetilde{\mathbb{Z}} \mapsto \widetilde{[-r, r]}$ définie par :

$$\forall n \in \widetilde{\mathbb{Z}}, \quad \beta_r^m(n) = \begin{cases} +\infty & \text{si } n > m + r, \\ -\infty & \text{si } n < m - r, \\ n - m & \text{sinon.} \end{cases}$$

Définition 2 Pour toute configuration $x \in \mathfrak{C}$, pour tous $r \in \mathbb{N}^*$ et $i \in \mathbb{Z}^d$, le cylindre de x de rayon r centré en i est la matrice $w = C_r^i(x)$ définie en dimension d par :

$$\forall k \in [-r, r]^d, \quad C_r^i(x)_k = \begin{cases} x_i & \text{si } k = 0, \\ \beta_r^0(x_{i+k}) & \text{si } k \neq 0 \text{ et } |x_i| = \infty, \\ \beta_r^{x_i}(x_{i+k}) & \text{sinon.} \end{cases}$$

Un cylindre w est donc un ensemble de mesures effectuées par un observateur, au centre duquel se trouve le point de référence $w_0 = x_i$ (voir figure 21).

Remarque 8 Lorsque le point de référence du cylindre contient un infini, l'observateur est placé à la hauteur 0. Si on ne le fait pas, les autres valeurs du cylindre ne peuvent pas être mesurées précisément (il faudrait une précision infinie), et la distance définie ci-dessous n'en serait pas une car on ne pourrait pas différencier certaines configurations pourtant différentes.

Définition 3 La distance entre deux configurations $x, y \in \mathfrak{C}$ est définie par $d(x, y) = 2^{-r}$, où $r = \min \{r \in \mathbb{N} \mid C_r^0(x) \neq C_r^0(y)\}$.

Proposition 38 La fonction d est une distance.

Preuve. Il est évident que pour tous $x, y \in \mathfrak{C}$, $d(x, y) = 0 \Leftrightarrow x = y$ et $d(x, y) = d(y, x)$. Enfin, remarquons que d est ultramétrique, c'est-à-dire que pour tous $x, y, z \in \mathfrak{C}$, $d(x, z) \leq \max(d(x, y), d(y, z))$. En effet, soit r le plus petit entier tel que $C_r^0(x) \neq C_r^0(y)$ et s le plus petit entier tel que $C_r^0(y) \neq C_r^0(z)$. Soit $t = \min(r, s)$, alors pour tout entier $u < t$ on a $C_u^0(x) = C_u^0(y) = C_u^0(z)$ et donc $d(x, z) \leq 2^{-t} = \max(d(x, y), d(y, z))$. \square

Avec la topologie induite sur \mathfrak{C} par la distance d , les cylindres permettent de former une base d'ouverts pour cette topologie : à chaque cylindre w correspond une *boule ouverte* $[w]_r = \{y \in \mathfrak{C} \mid C_r^0(y) = w\}$. En particulier, tous les \mathfrak{C}_u sont ouverts (boules de rayon 1). Quand il n'y a pas d'ambiguïté possible, on omet la précision r et les boules sont notées $[w]$. De manière générale, les ouverts de l'espace métrique \mathfrak{C} sont les ensembles de configurations ayant un nombre fini de valeurs fixées. On retrouve ici une topologie proche de la topologie de Cantor pour les automates cellulaires.

Dans la suite de la section on décrit toutes les propriétés topologiques de l'espace \mathfrak{C} .

Proposition 39 L'espace \mathfrak{C} est parfait (i.e. il n'a pas de points isolés).

Preuve. Soit $x \in \mathfrak{C}$ une configuration quelconque. Pour tout $r \in \mathbb{N}$, construisons une configuration x' égale à x partout sauf en un point $\mathbf{r} = (r + 1, 0, \dots, 0)$. On pose :

$$\forall i \in \mathbb{Z}^d \setminus \{\mathbf{r}\}, \quad x'_i = x_i \quad \text{et} \quad x'_{\mathbf{r}} = \begin{cases} 0 & \text{si } x_{\mathbf{r}} \neq 0, \\ 1 & \text{sinon.} \end{cases}$$

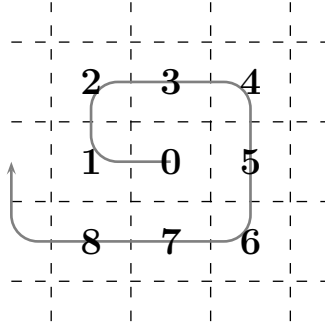
Par construction de x' , $0 < d(x, x') \leq 2^{-r-1}$. \square

La plupart des résultats sur la dynamique des systèmes dynamiques discrets repose sur le fait que l'espace des configurations est compact. Ici ce n'est malheureusement pas le cas, par exemple on ne peut extraire aucune sous-suite qui converge de la suite $(x^n)_{n \in \mathbb{N}}$ définie par $x_0^n = n$ et $x_i^n = 0$ pour $i \neq 0$ (tous les éléments de la suite sont à distance 1 les uns des autres). Par contre le corollaire 41 prouve que \mathfrak{C} est localement compact, en utilisant la proposition suivante qui montre que les ensembles \mathfrak{C}_u sont compacts.

Proposition 40 Pour tout $u \in \tilde{\mathbb{Z}}$, l'ensemble \mathfrak{C}_u est compact.

Preuve. Il faut commencer par trier les points de la grille en partant du centre, de manière à ce que leur norme (infinie) soit croissante. On utilise pour cela une bijection $\phi : \mathbb{N} \mapsto \mathbb{Z}^d$ telle que $i < j \Rightarrow |\phi(i)| \leq |\phi(j)|$. La figure 22 montre un exemple d'une telle fonction en dimension $d = 2$.

Soit $u \in \tilde{\mathbb{Z}}$, soit $E \in \mathfrak{C}_u$ un ensemble infini de configurations. Il suffit de construire une configuration y telle que pour tout $\varepsilon > 0$ il existe $x \in E$ tel que $d(x, y) < \varepsilon$. Nous construisons donc y petit à petit en remplissant au fur et à mesure tous les points de y dans l'ordre établi par ϕ , i.e. $\phi(0), \phi(1), \phi(2)$, etc. Soit $y_0 = u$, soit $U_0 = E$. On construit par induction à partir de U_0 une suite décroissante (pour l'inclusion) de sous-ensembles infinis de configurations de E . Pour tout $i \in \mathbb{N}^*$, considérons le multi-ensemble $\{x_{\phi(i)} \mid x \in U_{i-1}\}$.


 FIG. 22 – Exemple de bijection $\phi : \mathbb{N} \mapsto \mathbb{Z}^2$ croissante.

- (i) Soit il contient une valeur $k \in \tilde{\mathbb{Z}}$ qui revient infiniment souvent, on pose $y_{\phi(i)} = k$ et $U_i = \{x \in U_{i-1} \mid x_{\phi(i)} = k\}$.
- (ii) Soit il contient une suite strictement croissante $(k_j)_{j \in \mathbb{N}}$, posons $y_{\phi(i)} = +\infty$ et $U_i = \{x \in U_{i-1} \mid \exists j \in \mathbb{N}, x_{\phi(i)} = k_j\}$.
- (iii) Soit il contient une suite strictement décroissante $(k_j)_{j \in \mathbb{N}}$, posons $y_{\phi(i)} = -\infty$ et $U_i = \{x \in U_{i-1} \mid \exists j \in \mathbb{N}, x_{\phi(i)} = k_j\}$.

Soit $r > 0$, on cherche $x \in E$ tel que $d(x, y) < 2^{-r}$. Soit $n \in \mathbb{N}^*$ tel que $|\phi(n)| = r + 1$, considérons l'ensemble U_n . Pour tout $k \in \mathbb{Z}^d$ tel que $|k| \leq r$ on a soit $y_k = x_k$ pour tout $x \in U_n$ (si k a été construit avec le cas (i)), soit $y_k = +\infty$ [resp. $-\infty$] et U_n contient un nombre fini de configurations x telles que $x_k \leq r + x_0$ [resp. $x_k \geq -r + x_0$] (cas (ii) [resp. (iii)]). Il existe donc une infinité de configurations $x \in U_n \subset E$ telles que pour tout $|k| \leq r$, $y_k = +\infty \Rightarrow x_k - x_0 > r$, $y_k = -\infty \Rightarrow x_k - x_0 < -r$ et $|y_k| < \infty \Rightarrow x_k = y_k$. Pour n'importe laquelle de ces configurations x , $C_r^0(x) = C_r^0(y)$ et $d(x, y) < 2^{-r}$. \square

Corollaire 41 *L'espace \mathfrak{C} est localement compact (tout point de \mathfrak{C} a un voisinage compact) et donc complet.*

Preuve. Soit $x \in \mathfrak{C}$, on a $x \in \mathfrak{C}_{x_0}$ qui est compact et ouvert, c'est le voisinage cherché. \square

Corollaire 42 *Les boules ouvertes de \mathfrak{C} sont fermées.*

Preuve. Soit $[w]_r$ une boule ouverte de \mathfrak{C} . Soit

$$W = \left\{ v \in \widetilde{[-r, r]^{[-r, r]^d}} \mid v_0 = w_0, v \neq w \right\}$$

l'ensemble des cylindres différents de w et dont le point de référence est w_0 . On a $[w]_r = \mathfrak{C}_{w_0} \setminus \cup_{v \in W} [v]_r$. D'après la proposition 40, \mathfrak{C}_{w_0} est compact et donc fermé. La boule $[w]_r$ est donc fermée car c'est un fermé privé d'un certain nombre d'ouverts. \square

Corollaire 43 *L'espace \mathfrak{C} est totalement déconnecté.*

Preuve. Soient deux configurations distinctes $x, y \in \mathfrak{C}$ telles que $d(x, y) = 2^{-r} > 0$. Soit $B_x = [C_r^0(x)]_r$ la boule ouverte de centre x et de rayon 2^{-r} . B_x est fermée (corollaire 42) donc son complémentaire est ouvert. \mathfrak{C} est donc l'union disjointe de la boule ouverte B_x contenant x et de son complémentaire ouvert contenant y . \square

1.2 Automates de sable

Un automate de sable est un automate fini déterministe qui agit sur les configurations. Chaque point de la configuration d'entrée est mis à jour simultanément, grâce à une règle locale qui calcule la variation du nombre de grains d'un point en fonction du contenu de son voisinage (la *portée*). Le nombre de voisins pris en compte est appelé *rayon* de l'automate, il est toujours fini.

Définition 4 Pour toute configuration $x \in \mathfrak{C}$, pour tous $r \in \mathbb{N}^*$ et $i \in \mathbb{Z}^d$, la portée de x de rayon r centrée en i est la matrice de dimension d $R_r^i(x)$ définie par :

$$\forall k \in [-r, r]^d, \quad R_r^i(x)_k = \begin{cases} \perp & \text{si } k = 0, \\ \beta_r^{x_i}(x_{i+k}) & \text{sinon.} \end{cases}$$

En d'autres termes, une portée est un cylindre auquel on a enlevé le point de référence. Pour tout cylindre w , on note d'ailleurs $\langle\langle w \rangle\rangle$ la portée engendrée par w , i.e. $\langle\langle w \rangle\rangle_0 = \perp$ et pour tout $k \in [-r, r]^d \setminus \{0\}$, $\langle\langle w \rangle\rangle_k = w_k$. L'ensemble de toutes les portées de rayon r est noté \mathfrak{R}_r .

Définition 5 Un automate de sable (ou AS) est un triplet $\mathcal{A} = \langle d, r, f \rangle$ où $d \in \mathbb{N}^*$ est la dimension de l'automate, $r \in \mathbb{N}^*$ son rayon et $f : \mathfrak{R}_r \mapsto [-r, r]$ sa règle locale. À l'aide de la règle locale, on peut définir la règle globale $F : \mathfrak{C} \mapsto \mathfrak{C}$ par :

$$\forall x \in \mathfrak{C}, \forall i \in \mathbb{Z}^d, \quad F(x)_i = \begin{cases} x_i & \text{si } x_i = \pm\infty, \\ x_i + f(R_r^i(x)) & \text{sinon.} \end{cases}$$

La règle locale n'est rien d'autre qu'une *table de transitions* pouvant être décrite par un nombre fini de valeurs, ce qui fait des automates de sable un modèle intéressant du point de vue informatique car on peut facilement les implémenter. En l'absence d'ambiguïté, on assimile un automate \mathcal{A} à sa règle globale F .

Exemple 2 (automate \mathcal{S}) Cet automate simule le modèle SPM (voir partie I, chapitre 1) en mode synchrone. Il est défini par $\mathcal{S} = \langle 1, 1, f_{\mathcal{S}} \rangle$ avec :

$$\forall a, b \in \widetilde{[-1, 1]}, \quad f_{\mathcal{S}}(a, b) = \begin{cases} +1 & \text{si } a = +\infty \text{ et } b \neq -\infty, \\ -1 & \text{si } a \neq +\infty \text{ et } b = -\infty, \\ 0 & \text{sinon.} \end{cases}$$

Notons $F_{\mathcal{S}}$ la règle globale de \mathcal{S} . On retrouve le comportement de SPM : un grain tombe d'une colonne sur sa voisine de droite lorsque la différence de hauteur entre elles est strictement supérieure à 1 (voir figure 3, page 9). Tous les grains pouvant tomber le font en même temps car les automates de sable fonctionnent de manière synchrone, l'évolution de \mathcal{S} est donc celle représentée sur la figure 5(b) de la page 10.

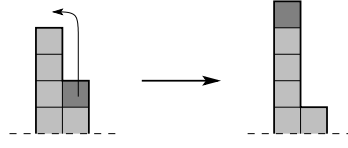
Exemple 3 (automate \mathcal{S}^r) Cet automate est défini de manière similaire à \mathcal{S} , la différence est que les grains remontent lorsqu'il y a une falaise (voir figure 23). Soit $\mathcal{S}^r = \langle 1, 1, f_{\mathcal{S}^r} \rangle$ avec :

$$\forall a, b \in \widetilde{[-1, 1]}, \quad f_{\mathcal{S}^r}(a, b) = \begin{cases} -1 & \text{si } a = +\infty \text{ et } b \neq -\infty, \\ +1 & \text{si } a \neq +\infty \text{ et } b = -\infty, \\ 0 & \text{sinon.} \end{cases}$$

La règle globale de \mathcal{S}^r est notée $F_{\mathcal{S}^r}$.

Remarque 9 L'automate \mathcal{S}^r est l'inverse à droite de \mathcal{S} .

D'autres exemples d'automates de sable, plus ou moins simples, seront décrits dans le chapitre suivant et en particulier dans la section 2.1.


 FIG. 23 – Fonctionnement de base de \mathcal{S}^r .

1.3 Relation avec les automates cellulaires

Les automates cellulaires [51, 50, 37] sont très souvent utilisés pour la modélisation de phénomènes régis par des lois locales, en raison de la simplicité de leur définition. La proposition 44 montre qu'on peut tout aussi bien utiliser les automates de sable.

Définition 6 Un automate cellulaire (ou AC) est un quadruplet $\langle S, d, r, g \rangle$ où S est un ensemble fini d'états, d la dimension de l'automate, r son rayon et $g : S^{[-r, r]^d} \mapsto S$ sa règle locale. La règle globale $G : S^{\mathbb{Z}^d} \mapsto S^{\mathbb{Z}^d}$ est définie par :

$$\forall x \in S^{\mathbb{Z}^d}, \forall i \in \mathbb{Z}^d, \quad G(x)_i = g(N_r^i(x)) ,$$

où $N_r^i(x)$ est le voisinage de x de rayon r centré en i , c'est la matrice définie par :

$$\forall x \in S^{\mathbb{Z}^d}, \forall i \in \mathbb{Z}^d, \forall j \in [-r, r]^d, \quad N_r^i(x)_j = x_{i+j} .$$

Tout automate cellulaire peut être simulé par un automate sur l'ensemble d'états $\{0, 1\}$, on se contentera donc de simuler les AC tels que $S = \{0, 1\}$.

Proposition 44 Tout $AC \mathcal{C} = \langle \{0, 1\}, d, r, g \rangle$ peut être simulé par un $AS \mathcal{A} = \langle d, 2r, f \rangle$.

Preuve. Soit $\mathcal{C} = \langle \{0, 1\}, d, r, g \rangle$ un AC quelconque de règle globale G , on prouve le résultat en supposant que $d = 1$ (les dimensions supérieures sont similaires). Toute configuration (d'automates cellulaires) $x \in \{0, 1\}^{\mathbb{Z}}$ est transformée en une configuration (d'automates de sable) $\phi(x) \in \mathfrak{C} = \tilde{\mathbb{Z}}^{\mathbb{Z}}$ telle que (voir figure 24) :

$$\forall i \in \mathbb{Z}, \quad \phi(x)_i = \begin{cases} x_{i/2} & \text{si } i \text{ est pair} , \\ 2 & \text{sinon.} \end{cases}$$

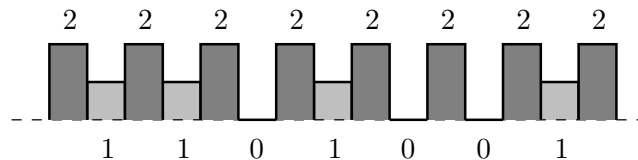


FIG. 24 – Un exemple de simulation d'AC par un AS. La configuration $(\dots, 1, 1, 0, 1, 0, 0, 1, \dots)$ est codée comme indiqué sur la figure, en intercalant une case sur deux un marqueur de hauteur 2.

La fonction ϕ est évidemment injective et donc toute configuration x peut être reconstruite à partir de $\phi(x)$. On simule \mathcal{C} par l'automate de sable $\mathcal{A} = \langle 1, 2r, f \rangle$, avec pour tout $w \in \mathfrak{R}_{2r}$:

$$f(w) = \begin{cases} g(w_{-2r}, w_{-2r+2}, \dots, w_{-2}, 0, w_2, \dots, w_{2r-2}, w_{2r}) & \text{si } w_{2i+1} = 2 \text{ pour } i \in [-r, r-1] , \\ g(w_{-2r} + 1, w_{-2r+2} + 1, \dots, w_{-2} + 1, 1, w_2 + 1, \dots, w_{2r-2} + 1, w_{2r} + 1) - 1 & \text{si } w_{2i+1} = 1 \text{ pour } i \in [-r, r-1] , \\ 0 & \text{sinon.} \end{cases}$$

Soit F la règle globale de \mathcal{A} et soit $x \in \{0, 1\}^{\mathbb{Z}}$. Évidemment, $F(\phi(x))_{2i+1} = 2$ pour $i \in \mathbb{Z}$, c'est le troisième cas de la règle locale f . Il reste à calculer l'image des indices pairs $2i$ pour $i \in \mathbb{Z}$. Soit $i \in \mathbb{Z}$, posons $w_j = \phi(x)_{2i+j} - \phi(x)_{2i}$ pour tout $j \in [-2r, 2r]$.

(i) Si $\phi(x)_{2i} = 0$; alors $x_i = 0$ et pour tout $j \in [-r, r-1]$, $w_{2j+1} = 2$ et :

$$\begin{aligned} F(\phi(x))_{2i} &= x_i + g(w_{-2r}, w_{-2r-2}, \dots, w_{2r-2}, w_{2r}) \\ &= g(\phi(x)_{2i-2r}, \phi(x)_{2i-2r+2}, \dots, \phi(x)_{2i+2r-2}, \phi(x)_{2i+2r}) \\ &= g(x_{i-r}, x_{i-r+1}, \dots, x_{i+r-1}, x_{i+r}) . \end{aligned}$$

(ii) Si $\phi(x)_{2i} = 1$; alors $x_i = 1$ et pour tout $j \in [-r, r-1]$, $w_{2j+1} = 2 - 1 = 1$, donc :

$$\begin{aligned} F(\phi(x))_{2i} &= x_i + g(w_{-2r} + 1, w_{-2r-2} + 1, \dots, w_{2r-2} + 1, w_{2r} + 1) - 1 \\ &= g(\phi(x)_{2i-2r}, \phi(x)_{2i-2r+2}, \dots, \phi(x)_{2i+2r-2}, \phi(x)_{2i+2r}) \\ &= g(x_{i-r}, x_{i-r+1}, \dots, x_{i+r-1}, x_{i+r}) . \end{aligned}$$

Dans tous les cas, $F(\phi(x)) = \phi(G(x))$ pour tout $x \in \{0, 1\}^{\mathbb{Z}}$. □

Les deux modèles ont la même puissance de calcul, comme le montre la proposition suivante (noter que pour simuler un AS avec un AC, il faut augmenter la dimension).

Proposition 45 *Tout AS $\mathcal{A} = \langle d, r, f \rangle$ peut être simulé par un AC $\mathcal{C} = \langle \{0, 1\}, d+1, 2r, g \rangle$.*

Preuve. Soit $\mathcal{A} = \langle d, r, f \rangle$ un AS. Cette fois encore, nous donnons la preuve pour $d = 1$, elle est similaire pour les autres valeurs. Une configuration $x \in \mathfrak{C} = \tilde{\mathbb{Z}}^{\mathbb{Z}}$ est codée par $\psi(x) \in \{0, 1\}^{\mathbb{Z}^2}$ définie par (voir figure 25) :

$$\forall i, j \in \mathbb{Z}, \quad \psi(x)_{i,j} = \begin{cases} 1 & \text{si } x_i > j , \\ 0 & \text{sinon.} \end{cases}$$

Encore une fois, ψ étant injective il est possible de reconstruire la configuration x de départ à partir de son image $\psi(x)$.

Soit $\langle\langle w \rangle\rangle = (w_{-r}, \dots, w_{-1}, \perp, w_1, \dots, w_r)$ une portée. Pour simplifier les expressions on pose $\langle\langle w \rangle\rangle_0 = 0$ au lieu de \perp . Posons également $n = f(\langle\langle w \rangle\rangle)$, on a $n \in [-r, r]$. Soit $\mathcal{C} = \langle \{0, 1\}, d+1, 2r, g \rangle$ l'AC dont la règle locale g est définie dans chacun des cas suivants.

(i) Si $n > 0$; pour tout $k \in [0, n-1]$, pour tous les voisinages N de rayon $r+n$ tels que :

$$\forall i \in [-r, r], \forall j \in [-k-r-1, -k+r], \quad N_{i,j} = \begin{cases} 0 & \text{si } j \geq w_i - k , \\ 1 & \text{sinon,} \end{cases}$$

on définit $g(N) = 1$.

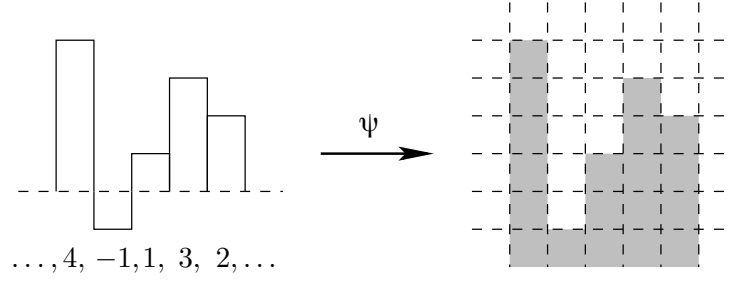


FIG. 25 – Simulation d'un AS par un AC. La configuration $(\dots, 4, -1, 1, 3, 2, \dots)$ est « aplatie » pour être codée en dimension 2. Dans $\psi(x)$, les cases sont grisées pour la valeur 1, vides pour 0.

(ii) Si $n < 0$; pour tout $k \in [n, -1]$, pour tous les voisinages N de rayon $r + |n|$ tels que :

$$\forall i \in [-r, r], \forall j \in [-k - r - 1, -k + r], \quad N_{i,j} = \begin{cases} 0 & \text{si } j \geq w_i - k, \\ 1 & \text{sinon,} \end{cases}$$

on définit $g(N) = 0$.

(iii) Dans tous les autres cas, g laisse inchangée la valeur de la cellule centrale.

Si l'on appelle G la règle locale, on a par construction de g que $G(\psi(x)) = \psi(F(x))$ pour tout $x \in \mathfrak{C}$. \square

L'exemple suivant illustre la simulation décrite dans la preuve de la proposition 45.

Exemple 4 L'automate de sable \mathcal{S} défini dans l'exemple 2 peut être simulé simplement par l'automate cellulaire $\mathcal{C} = \langle \{0, 1\}, 2, 2, g_{\mathcal{S}} \rangle$. Le cas où la règle locale $f_{\mathcal{S}}$ renvoie -1 est exprimé par les règles :

$$g_{\mathcal{S}} \begin{pmatrix} 0 & 0 & 0 \\ \alpha & 0 & 0 \\ \beta & \mathbf{1} & 0 \\ \gamma & 1 & 0 \\ \delta & 1 & \lambda \end{pmatrix} = 0,$$

pour $0 \leq \alpha \leq \beta \leq \gamma \leq \delta \leq 1$ et $0 \leq \lambda \leq 1$. Le $\mathbf{1}$ en gras indique le centre du voisinage. De même, $f_{\mathcal{S}}$ retourne $+1$ se traduit par :

$$g_{\mathcal{S}} \begin{pmatrix} \alpha & 0 & \beta \\ 1 & 0 & \gamma \\ 1 & \mathbf{0} & \delta \\ 1 & 1 & \lambda \\ 1 & 1 & 1 \end{pmatrix} = 1,$$

pour $0 \leq \alpha \leq 1$ et $0 \leq \beta \leq \gamma \leq \delta \leq \lambda \leq 1$. Pour tout autre voisinage N en entrée, $g_{\mathcal{S}}(N)$ ne modifie pas la valeur centrale.

Remarque 10 Automates de sable et automates cellulaires peuvent simuler les mêmes choses, mais les preuves des deux propositions de cette partie donnent déjà un premier aperçu des avantages et des inconvénients de chacun. Les AC peuvent mesurer les valeurs exactes de leur voisinage, alors que pour simuler un tel comportement avec les AS il faut augmenter le rayon (proposition 44).

D'un autre côté, les AS sont des AC « sans trous » (figure 25, pas de vide entre deux cases grisées de même abscisse) et leur ensemble infini d'états économise une dimension (proposition 45) pour des calculs sur un nombre d'états arbitraire.

1.4 Caractérisation « à la Hedlund »

Nous prouvons dans cette partie l'équivalent du théorème de Hedlund [32] pour les automates cellulaires. Cela permet de faire le lien entre le côté informatique des automates de sable, à travers la description finie de leur table de transitions, et le point de vue mathématique lié à la règle locale et à la notion de système dynamique discret.

Ce résultat de représentation fort (théorème 47) caractérise une classe de fonctions de \mathfrak{C} dans \mathfrak{C} qui ont une description finie. De telles fonctions sont intéressantes du point de vue informatique car elles peuvent être implémentées de manière exacte dans des machines à mémoire finie, et donc permettent de faire des simulations précises, sans risque d'amplifier des erreurs dues aux approximations initiales.

Ce théorème permet ensuite de montrer que si un automate de sable est réversible (*i.e.* sa règle globale est bijective), son inverse est également un automate de sable.

Pour une dimension d donnée et pour tout entier $k \in [0, d-1]$, notons $\mathbb{1}_k$ le vecteur dont toutes les coordonnées sont nulles sauf la k^e qui vaut 1. Le k^e *décalage horizontal* $\sigma_k : \mathfrak{C} \mapsto \mathfrak{C}$ est défini par :

$$\forall x \in \mathfrak{C}, \forall i \in \mathbb{Z}^d, \quad \sigma_k(x)_i = x_{i+\mathbb{1}_k} .$$

En dimension 1, il n'y a qu'un seul décalage que l'on note simplement σ . Le *décalage vertical* $\rho : \mathfrak{C} \mapsto \mathfrak{C}$ est défini par :

$$\forall x \in \mathfrak{C}, \forall i \in \mathbb{Z}^d, \quad \rho(x)_i = x_i + 1 .$$

Une fonction $F : \mathfrak{C} \mapsto \mathfrak{C}$ est dite *invariante horizontalement* [resp. *invariante verticalement*] si pour tout $k \in [0, d-1]$, $F \circ \sigma_k = \sigma_k \circ F$ [resp. $F \circ \rho = \rho \circ F$]. Enfin, $F : \mathfrak{C} \mapsto \mathfrak{C}$ *conserve les infinis* si :

$$\forall x \in \mathfrak{C}, \forall i \in \mathbb{Z}^d, \quad \begin{cases} F(x)_i = +\infty \Leftrightarrow x_i = +\infty \\ \text{et} \\ F(x)_i = -\infty \Leftrightarrow x_i = -\infty . \end{cases}$$

Le théorème de représentation des automates de sable nécessite un lemme préliminaire.

Lemme 46 *Pour toute fonction $F : \mathfrak{C} \mapsto \mathfrak{C}$ continue, invariante verticalement et qui conserve les infinis, pour tout $u \in \tilde{\mathbb{Z}}$, $F^{-1}(\mathfrak{C}_u)$ est compact.*

Preuve. Soit F une fonction continue de \mathfrak{C} dans \mathfrak{C} , invariante verticalement et qui conserve les infinis. Si $|u| = \infty$, $F^{-1}(\mathfrak{C}_u) \subseteq \mathfrak{C}_u$ car F conserve les infinis. Comme F est continue et que \mathfrak{C}_u est compact (proposition 40), $F^{-1}(\mathfrak{C}_u)$ est fermé et donc compact.

Si u est fini, soient $U = f^{-1}(\mathfrak{C}_u)$ (U est fermé), $U_i = U \cap \mathfrak{C}_i$ pour $i \in \tilde{\mathbb{Z}}$ et $I = \{i \in \tilde{\mathbb{Z}} \mid U_i \neq \emptyset\}$. On prouve par l'absurde que I contient un nombre fini d'éléments. Supposons que $|I| = \infty$, pour tout $i \in I$ on prend $x^i \in \rho^{-i}(U_i)$ en notant que $x^i \in \mathfrak{C}_0$. Comme \mathfrak{C}_0 est compact, on peut extraire de la suite $(x^i)_{i \in I}$ une sous-suite $(x^{\phi(n)})_{n \in \mathbb{N}}$ qui converge vers $y \in \mathfrak{C}_0$. Comme F est continue, $\lim_{n \rightarrow \infty} F(x^{\phi(n)}) = F(y)$ et donc si $v \in \tilde{\mathbb{Z}}$ est tel que $F(y) \in \mathfrak{C}_v$, il y a un entier N tel que pour tout $n \geq N$, $F(x^{\phi(n)}) \in \mathfrak{C}_v$.

Soient $n_1 = \phi(N)$ et $n_2 = \phi(N+1)$. On remarque que pour a et b quelconques, si $\mathfrak{C}_a \cap \mathfrak{C}_b \neq \emptyset$ alors $a = b$. Comme F est invariante verticalement, $F(\rho^{n_1}(x_{n_1})) \in \mathfrak{C}_{v+n_1} \cap \mathfrak{C}_u$ et $F(\rho^{n_2}(x_{n_2})) \in$

$\mathfrak{C}_{v+n_2} \cap \mathfrak{C}_u$, d'où $n_1 = n_2$ ce qui est impossible puisque ϕ est injective. $|\mathbf{I}|$ est fonc fini, et comme $U \subset \cup_{i \in \mathbf{I}} \mathfrak{C}_i$, U est un fermé dans un compact et donc est compact. \square

Théorème 47 *Une fonction $F : \mathfrak{C} \mapsto \mathfrak{C}$ est la règle globale d'un automate de sable si et seulement si*

- (i) F est continue ;
- (ii) F est invariante horizontalement ;
- (iii) F est invariante verticalement ;
- (iv) F conserve les infinis.

Preuve. Soit $\mathcal{A} = \langle d, r, f \rangle$ un automate de sable de règle globale F . Par définition de \mathcal{A} , F est invariante par les décalages horizontal et vertical, et conserve les infinis. Montrons que F est continue. Soit une configuration $x \in \mathfrak{C}$ et un entier $l \in \mathbb{N}$. Il faut trouver un entier m tel que pour toute configuration $y \in \mathfrak{C}$, $d(x, y) < 2^{-m}$ implique $d(F(x), F(y)) < 2^{-l}$. On pose $m = 3r + l$, soit y tel que $d(x, y) < 2^{-m}$. On a donc $C_m^0(x) = C_m^0(y)$ et $x_0 = y_0$ (car $m > 0$), donc $F(x)_0 = F(y)_0$ d'une part, et $\beta_m^{x_0}(x_i) = \beta_m^{y_0}(y_i)$ pour tout $i \in [-m, m]^d$ d'autre part. Pour tout $j \in [-l, l]^d \setminus \{0\}$ il y a deux cas à distinguer.

(i) Si $|x_j - x_0| > 2r + l$; alors comme $x_0 = y_0$ et $\beta_m^{x_0}(x_j) = \beta_m^{y_0}(y_j)$ on a $|y_j - y_0| > 2r + l$. Alors $|F(x)_j - F(x)_0| \geq |x_j - x_0| - |F(x)_0 - x_0| - |F(x)_j - x_j| > (2r + l) - r - r = l$ et de la même façon $|F(y)_j - F(y)_0| > l$.

(ii) Si $|x_j - x_0| \leq 2r + l < m$; alors $x_j = y_j$ puisque $x_0 = y_0$ et $\beta_m^{x_0}(x_j) = \beta_m^{y_0}(y_j)$. Pour tout vecteur $k \in [j-r, j+r]^d \setminus \{0\}$, il y a trois cas possibles (toujours puisque $\beta_m^{x_0}(x_j) = \beta_m^{y_0}(y_j)$) :

- si $x_k = y_k$; donc $\beta_r^{x_j}(x_k) = \beta_r^{y_j}(y_k)$;
- si $x_k - x_0 > m = 3r + l$; alors $y_k - y_0 > m = 3r + l$. Comme $|x_j - x_0| \leq 2r + l$ et $|y_j - y_0| \leq 2r + l$ (car $x_0 = y_0$ et $x_j = y_j$), $x_k - x_j > r$ et $y_k - y_j > r$;
- si $x_k - x_0 < -m = -3r - l$; en faisant comme pour le cas précédent on trouve que $x_k - x_j < -r$ et $y_k - y_j < -r$.

Dans tous les cas on conclut que $\beta_r^{x_j}(x_k) = \beta_r^{y_j}(y_k)$. Comme $x_j = y_j$, il s'ensuit que $F(x)_j = F(y)_j$.

À la fois pour (i) et pour (ii) on a donc $\beta_l^{F(x)_0}(F(x)_j) = \beta_l^{F(y)_0}(F(y)_j)$ pour tout $j \in [-l, l]^d \setminus \{0\}$, par conséquent $C_l^0(F(x)) = C_l^0(F(y))$ et $d(F(x), F(y)) < 2^{-l}$.

Pour la réciproque, considérons une fonction $F : \mathfrak{C} \mapsto \mathfrak{C}$ continue, invariante verticalement et horizontalement et qui conserve les infinis. Soit $U = F^{-1}(\mathfrak{C}_0)$, U est compact d'après le lemme 46 et est donc l'union d'un nombre fini de boules ouvertes : $U = \cup_{i \in \mathbf{I}} [w^i]_{r_i}$ avec $|\mathbf{I}| < \infty$. On peut se ramener à des boules de même rayon r (une boule de rayon r est l'union d'un nombre fini de boules de rayon quelconque $r_i < r$), soit après renommage : $U = \cup_{i \in \mathbf{I}} [w^i]_r = \cup_{i \in \mathbf{I}} [w^i]$ en ne notant plus le rayon des boules. Montrons que pour tous $i, j \in \mathbf{I}$, $i \neq j$, les portées $\langle\langle w^i \rangle\rangle$ et $\langle\langle w^j \rangle\rangle$ sont différentes. Supposons qu'il existe $i \neq j$ tels que $\langle\langle w^i \rangle\rangle = \langle\langle w^j \rangle\rangle$. Appelons a et b les points de référence des deux cylindres, soit $a = w_0^i$ et $b = w_0^j$. Comme les cylindres w^i et w^j sont différents mais ont même portée, il est évident que $a \neq b$. Soit $x \in [w^i]$ et $y = \rho^{b-a}(x) \in [w^j]$. Comme F est invariante verticalement, $F(y) = \rho^{b-a}(F(x)) \in \mathfrak{C}_{b-a}$, mais on a également $F(y) \in \mathfrak{C}_0$ ce qui est impossible car $\mathfrak{C}_{b-a} \cap \mathfrak{C}_0 = \emptyset$. Toutes les portées $\langle\langle w^i \rangle\rangle$ sont donc différentes.

Montrons maintenant que la suite $(\langle\langle w^i \rangle\rangle)_{i \in \mathbf{I}}$ contient toutes les portées de rayon r possibles. Supposons qu'il existe une configuration x telle que $x_0 = 0$ et $R_r^0(x) \neq \langle\langle w^i \rangle\rangle$ pour tout $i \in \mathbf{I}$. On a

$F(x) \in \mathfrak{C}_u$ pour un certain $u \in \mathbb{Z}$ ($|u| \neq \infty$ car F conserve les infinis), donc par invariance verticale de F on obtient $F(\rho^{-u}(x)) \in \mathfrak{C}_0$ et donc $R_r^0(\rho^{-u}(x)) \in (\langle\langle w^i \rangle\rangle)_{i \in I}$. Or $R_r^0(\rho^{-u}(x)) = R_r^0(x)$, il y a contradiction et donc toutes les portées apparaissent bien dans la suite $(\langle\langle w^i \rangle\rangle)_{i \in I}$.

Il est donc naturel de définir la fonction $f : \mathfrak{R}_r \mapsto [-r, r]$ par $f(\langle\langle w^i \rangle\rangle) = -w_0^i$. Soit F' la règle globale de l'automate de sable $\langle d, r, f \rangle$, montrons que $F' = F$. Soient une configuration $x \in \mathfrak{C}$ et un vecteur $n \in \mathbb{Z}^d$, soit $i \in I$ tel que $\langle\langle w^i \rangle\rangle = R_r^n(x)$. On a $F'(x)_n = x_n - w_0^i$. Comme F est invariante horizontalement et verticalement, on a aussi :

$$F(x)_n = \sigma^n(F(x))_0 = F(\sigma^n(x))_0 = F(\rho^{w_0^i - x_n}(\sigma^n(x)))_0 + x_n - w_0^i.$$

Soit $y = \rho^{w_0^i - x_n}(\sigma^n(x))$, on a $C_r^0(y) = w^i$ et donc par définition de w^i , $F(y)_0 = 0$. On obtient donc $F(x)_n = x_n - w_0^i = F'(x)_n$, soit $F' = F$. \square

Ce théorème permet de montrer un dernier résultat important et intéressant. À l'aide d'un nouveau lemme, on montre que l'inverse d'un automate de sable réversible est encore un automate de sable.

Lemme 48 *Pour tout AS de règle globale F , si F est injective alors F est ouverte.*

Preuve. Soit un AS $\langle d, r, f \rangle$ de règle globale F injective. Soit A une boule ouverte (et fermée, corollaire 42) $[w]_l$, $l \geq 0$. Par définition des AS, $F(A) \subset \cup_{i \in [w_0 - r, w_0 + r]} \mathfrak{C}_i$ qui est une union finie d'ouverts compacts (proposition 40). Soit $C = F^{-1}(\cup_{i \in [w_0 - r, w_0 + r]} \mathfrak{C}_i)$ et $B = C \setminus A$. Comme F est continue, C est un ouvert fermé, or A l'est également donc B est aussi ouvert fermé. Toujours par définition de l'automate, $C \subset \cup_{i \in [w_0 - 2r, w_0 + 2r]} \mathfrak{C}_i$ donc C est compact en tant que fermé dans un compact. $B \subset C$ est donc lui aussi compact, $F(B)$ aussi car F est continue. Puisque F est injective et que $A = C \setminus B$, $F(A) = F(C) \setminus F(B)$. $F(C)$ est ouvert, $F(B)$ est fermé donc finalement $F(A)$ est ouvert. \square

Proposition 49 *Pour tout AS de règle globale F , si F est bijective alors F^{-1} est la règle globale d'un AS.*

Preuve. Soit \mathcal{A} un AS de règle globale F . D'après le lemme 48, F est ouverte et donc F^{-1} est continue. Pour toute configuration $x \in \mathfrak{C}$ on pose $y = F^{-1}(x)$. Puisque F est invariante verticalement, $F(\rho(x)) = \rho(f(x))$ d'où $F(\rho(F^{-1}(y)) = \rho(y)$ et en composant avec F^{-1} à gauche, $\rho(F^{-1}(y)) = F^{-1}(\rho(y))$ donc F^{-1} est invariante verticalement. Il en est de même avec σ donc F^{-1} est invariante horizontalement. Clairement, F^{-1} conserve les infinis donc d'après le théorème 47, F^{-1} est bien la règle globale d'un AS. \square

Dynamique des automates de sable

Sommaire

2.1	Propriétés ensemblistes	60
2.2	Automates conservateurs de grains	68
2.2.1	Définitions	69
2.2.2	Décidabilité	71
2.3	Automates ultimement périodiques et ultimement stables	74
2.3.1	Construction de l'automate de sable	74
2.3.2	Simulation de la machine à deux compteurs	76
2.3.3	Arrêt sur erreurs	79
2.3.4	Décidabilité	81

Dans ce chapitre, nous nous intéressons aux propriétés concernant la dynamique des automates de sable. Nous cherchons à évaluer dans quelle mesure la table de transitions (quantité d'informations finie) détermine le comportement à long terme de l'automate. L'étude des comportements qui nous intéressent est souvent difficile dans le cadre général des systèmes dynamiques discrets, cela se confirme ici : la plupart sont indécidables.

Certaines de ces propriétés permettent d'évaluer le degré de *chaoticité* d'un automate. Cela va de l'équicontinuité pour les automates les plus stables à l'expansivité, en passant par la sensibilité aux conditions initiales et la quasi-équicontinuité (voir la classification pour les automates cellulaires introduite dans [36]). L'expansivité, mais aussi la transitivité et la régularité dépendent du fait qu'un automate soit surjectif (et non injectif pour les expansifs). Pour cette raison nous commençons par étudier les relations existant entre l'injectivité et la surjectivité (partie 2.1, figure 31), sans toutefois pouvoir conclure sur leur décidabilité.

De plus, comme nous voulons que notre modèle puisse trouver des applications dans la simulation des phénomènes naturels, il est important de vérifier s'il en possède les caractéristiques essentielles. Par exemple, pour simuler un système de piles de sable, il faut savoir si un automate déplace effectivement des grains, plutôt que de les détruire ou d'en créer. Nous définissons donc une classe d'automates de sable, appelés conservateurs de grains, qui ne modifient en aucun cas le nombre de grains présents dans la configuration initiale. Dans la partie 2.2, nous prouvons que l'appartenance à cette classe est décidable (corollaire 66).

Pour terminer, nous cherchons à identifier les automates au comportement « simple » : automates ultimement périodiques (fonctionnement périodique au bout d'un nombre fini d'étapes),

ultimement stables (les configurations ne sont plus modifiées après un nombre fini d'étapes), nilpotents (toutes les configurations arrivent sur une configuration constante au bout d'un nombre fini d'itérations). Le cas de la nilpotence est difficile, car il faut commencer par trouver une définition qui capture le comportement désiré et qui contient un nombre « raisonnable » d'automates. Si l'on s'inspire des automates cellulaires [33, 16, 34], un automate de sable est nilpotent si son ensemble limite ne contient que des configurations uniformes. On n'a aucun exemple d'automate de sable vérifiant cette propriété, il est probable qu'il n'en existe pas. Intuitivement, l'automate \mathcal{L} (défini page 62) devrait être nilpotent, car il envoie toutes ses colonnes en direction leur voisine de gauche. En particulier, si l'on part d'une configuration finie, toutes les colonnes contiendront 0 au bout d'un temps fini. Le problème est qu'une « vague » se propage vers la droite, à l'infini, et qu'une configuration constante n'est atteinte qu'au bout d'un temps infini. En fait, \mathcal{L} est surjectif (proposition 53) donc son ensemble limite contient \mathfrak{C} tout entier. Les deux autres propriétés, stabilité et périodicité ultimes, sont indécidables (partie 2.3, théorèmes 71 et 72), ce qui renforce l'idée que la dynamique des automates de sable est complexe.

2.1 Propriétés ensemblistes

Dans cette partie nous commençons l'étude de notre modèle par des propriétés ensemblistes simples, injectivité et surjectivité. Dans la théorie des systèmes dynamiques, elles permettent d'évaluer le degré de chaoticité d'un automate. Ces propriétés de réversibilité sont également utiles du point de vue de la simulation, surtout que l'on sait que l'inverse d'un automate de sable est encore un automate de sable (proposition 49). En effet, il est intéressant de pouvoir reconstruire l'origine d'un phénomène naturel (une avalanche par exemple), et de pouvoir s'assurer qu'à partir d'une origine différente on produirait un autre résultat.

Nous commençons par montrer quelques relations basiques entre différentes variantes de ces propriétés, dans l'esprit de ce qui a été fait dans [19] pour les automates cellulaires. Elles ont été publiées dans [9, 10]. Ces résultats permettent de mieux comprendre le modèle, avant de passer à des propriétés dynamiques plus complexes.

Quelques définitions sont encore nécessaires. Une configuration $x \in \mathfrak{C}$ est dite *finie* s'il existe $k \in \mathbb{N}$ tel que pour tout vecteur $i \in \mathbb{Z}^d$, $|i| \geq k \Rightarrow x_i = 0$ et $|i| < k \Rightarrow |x_i| < \infty$. Une configuration finie contient donc un nombre fini d'éléments non nuls (ou par extension non constants, c'est équivalent en raison de l'invariance verticale d'un AS) et aucune valeur infinie. La *taille* d'une configuration x finie est $|x| = \max_{i,j \in \mathbb{Z}^d} \{|i - j|, x_i \neq 0 \text{ et } x_j \neq 0\}$. L'ensemble des configurations finies est noté \mathfrak{F} .

Une configuration $x \in \mathfrak{C}$ est dite *périodique* (sous-entendu spatialement) s'il existe un vecteur $p = (p_1, \dots, p_d) \in \mathbb{N}^{*d}$ (appelé *période*) tel que pour tout vecteur $i = (i_1, \dots, i_d) \in \mathbb{Z}^d$ et pour tous les entiers $t_1, \dots, t_d \in \mathbb{Z}$, $x_i = x_{i_1+t_1p_1, \dots, i_d+t_dp_d}$ et $|x_i| < \infty$. Une configuration périodique est une configuration périodique selon chaque direction, dont aucune valeur n'est infinie. L'ensemble des configurations (spatialement) périodiques est noté \mathfrak{P} .

On laisse la possibilité aux configurations finies et périodiques de contenir des infinis. Prenons $\tilde{\mathfrak{F}}$ défini par $x \in \tilde{\mathfrak{F}}$ si et seulement s'il existe $k \in \mathbb{N}$ tel que pour tout $i \in \mathbb{Z}^d$, $|i| \geq k \Rightarrow x_i = 0$. De manière similaire, on dit que $x \in \tilde{\mathfrak{P}}$ si et seulement s'il existe $p \in \mathbb{N}^{*d}$ tel que pour tous $i, t \in \mathbb{Z}^d$ on ait $x_i = x_{i_1+t_1p_1, \dots, i_d+t_dp_d}$.

Pour finir, un automate de sable \mathcal{A} de règle globale F est *injectif* [resp. *surjectif*] si F est injective [resp. surjective]. Pour tout ensemble $\mathfrak{U} \subseteq \mathfrak{C}$, \mathcal{A} et F sont \mathfrak{U} -injectifs [resp. \mathfrak{U} -surjectifs] si la restriction de F à \mathfrak{U} est injective [resp. surjective].

La proposition suivante permet de n'étudier que les ensembles \mathfrak{F} et \mathfrak{P} , toutes les propriétés étudiées étant équivalentes avec ou sans infinis.

Proposition 50 *Un AS est \mathfrak{F} -injectif [resp. \mathfrak{F} -surjectif] si et seulement si il est $\tilde{\mathfrak{F}}$ -injectif [resp. $\tilde{\mathfrak{F}}$ -surjectif], et \mathfrak{P} -injectif [resp. \mathfrak{P} -surjectif] si et seulement si il est $\tilde{\mathfrak{P}}$ -injectif [resp. $\tilde{\mathfrak{P}}$ -surjectif].*

Preuve. On ne prouve ces équivalences que sur \mathfrak{F} et $\tilde{\mathfrak{F}}$, elles se démontrent de manière similaire sur \mathfrak{P} et $\tilde{\mathfrak{P}}$. Soit F la règle globale d'un automate $\tilde{\mathfrak{F}}$ -surjectif, montrons que F est \mathfrak{F} -surjective. Soit $x \in \mathfrak{F} \subset \tilde{\mathfrak{F}}$. Alors il existe $y \in \tilde{\mathfrak{F}}$ tel que $F(y) = x$. Comme il n'y a pas de colonnes infinies dans x et que F conserve les infinis, $y \in \mathfrak{F}$.

Réciproquement, soit F la règle globale d'un automate \mathfrak{F} -surjectif, soit $x \in \tilde{\mathfrak{F}}$. Si $x \in \mathfrak{F}$ alors ses préimages sont dans \mathfrak{F} . Sinon, soit $x' \in \mathfrak{F}$ définie par :

$$\forall i \in \mathbb{Z}^d, \quad x'_i = \begin{cases} x_i & \text{si } |x_i| < \infty, \\ M + 3r + 1 & \text{si } x_i = +\infty, \\ m - 3r - 1 & \text{si } x_i = -\infty, \end{cases}$$

où $M = \max_{i \in \mathbb{Z}^d} \{x_i, |x_i| < \infty\}$ et $m = \min_{i \in \mathbb{Z}^d} \{x_i, |x_i| < \infty\}$ sont respectivement la plus grande et la plus petite valeur finie prise par x . Soient $y' \in \mathfrak{F}$ tel que $y' = f(x')$, et $y \in \tilde{\mathfrak{F}}$ défini par :

$$\forall i \in \mathbb{Z}^d, \quad y_i = \begin{cases} y'_i & \text{si } |x_i| < \infty, \\ +\infty & \text{si } x_i = +\infty, \\ -\infty & \text{si } x_i = -\infty. \end{cases}$$

Alors, il est toujours vrai que :

$$\forall i \in \mathbb{Z}^d, \quad F(y)_i = \begin{cases} F(y'_i) = x'_i = x_i & \text{si } |x_i| < \infty, \\ +\infty & \text{si } x_i = +\infty, \\ -\infty & \text{si } x_i = -\infty. \end{cases}$$

Seule l'égalité $F(y)_i = F(y')_i$ n'est pas évidente. Elle vient du fait que les cylindres de rayon r centrés en i pour y et pour y' sont identiques, par construction de y et y' . En effet, dans le cas $|x_i| < \infty$, $y_i = y'_i$ et pour tout $j \in [-r, r]^d$, si $|y_{i+j}| < \infty$ alors $y_{i+j} = y'_{i+j}$ (cas $|x_{i+j}| < \infty$). Si $y_{i+j} = +\infty$ alors $y'_{i+j} - y'_i = F(y')_{i+j} - [F(y')_{i+j} - y'_{i+j}] - [y'_i - F(y')_i] - F(y')_i = x'_{i+j} - [F(y')_{i+j} - y'_{i+j}] - [y'_i - F(y')_i] - x'_i \geq [M + 3r + 1] - r - r - M = r + 1 > r$. On procède de même si $y_{i+j} = -\infty$, et donc dans tous les cas, $C_r^i(y) = C_r^i(y')$ et $F(y)_i = F(y')_i$. On a bien $F(y) = x$, F est $\tilde{\mathfrak{F}}$ -surjective.

Soit F la règle globale d'un automate $\tilde{\mathfrak{F}}$ -injectif. Soient $x^1, x^2 \in \mathfrak{F} \subset \tilde{\mathfrak{F}}$, avec $x^1 \neq x^2$. Comme F est $\tilde{\mathfrak{F}}$ -injective, $F(x^1) \neq F(x^2)$.

Réciproquement, soit F la règle globale d'un automate \mathfrak{F} -injectif, soient $x^1, x^2 \in \tilde{\mathfrak{F}}$. Comme précédemment, les colonnes infinies de x^1 et x^2 sont remplacées par des colonnes de hauteur $M + r + 1$ ou $m + r + 1$ ($M = \min(M^1, M^2)$ et $m = \min(m^1, m^2)$ sont les valeurs maximales et minimales atteintes par x^1 et x^2), cela donne de nouvelles configurations $x^{1'}, x^{2'} \in \mathfrak{F}$ avec $x^{1'} \neq x^{2'}$. Il existe un indice $i \in \mathbb{Z}^d$ tel que $F(x^{1'})_i \neq F(x^{2'})_i$ car F est \mathfrak{F} -injective, il faut considérer les cas suivants :

- (i) si $|x_i^1| < \infty$ et $|x_i^2| < \infty$; alors $F(x^1)_i = F(x^{1'})_i \neq F(x^{2'})_i = F(x^2)_i$ car $C_r^i(x^1) = C_r^i(x^{1'})$ et $C_r^i(x^2) = C_r^i(x^{2'})$;
- (ii) si $x_i^1 = +\infty$; alors forcément $x_i^2 \neq +\infty$, sinon $x^{1'} = x^{2'} = M + r + 1$ et $F(x^{1'})_i = F(x^{2'})_i$ (les portées contiennent $-\infty$ partout), d'où $F(x^1)_i = +\infty \neq F(x^2)_i$;

(iii) si $x_i^1 = -\infty$ ou $x_i^2 = \pm\infty$; une étude semblable à celle du cas précédent conduit à $F(x^1)_i \neq F(x^2)_i$.

Dans tous les cas, $F(x^1) \neq F(x^2)$ donc F est \mathfrak{F} -injective. \square

Proposition 51 *Tout automate de sable \mathfrak{P} -surjectif est surjectif.*

Preuve. Soit un automate de sable \mathfrak{P} -surjectif de règle globale F et de rayon r , soit une configuration $x \in \mathfrak{C}$. Pour tout $n \in \mathbb{N}$, on définit $x^n \in \mathfrak{P}$ comme la configuration périodique de période $(2n+1, \dots, 2n+1)$ qui vérifie que pour tout $i \in \mathbb{Z}^d$, $|i| \leq n \Rightarrow x_i^n = x_i$. Soit $y^n \in \mathfrak{C}$ tel que $x^n = F(y^n)$ (F est \mathfrak{P} -surjective donc \mathfrak{P} -surjective d'après la proposition 50). Chaque y^n appartient à un \mathfrak{C}_u avec $u \in U \subseteq [x_0 - r, x_0 + r]$, d'après la définition de l'automate. La suite $(y^n)_{n \in \mathbb{N}}$ est contenue dans le compact (proposition 40, page 50) $\cup_{u \in U} \mathfrak{C}_u$. On peut donc en extraire une sous-suite convergente $(y^{n_k})_{k \in \mathbb{N}}$ avec $y = \lim_{k \rightarrow +\infty} y^{n_k}$. Par contradiction on suppose que $F(y) \neq x$, c'est-à-dire qu'il existe $i \in \mathbb{Z}^d$ tel que $F(y)_i \neq x_i$. C'est impossible car pour tout k suffisamment grand (il suffit que $n_k \geq |i|$), on a $F(y)_i = F(y^{n_k})_i = x_i^{n_k} = x_i$. La configuration $y \in \mathfrak{C}$ est donc une préimage de x . \square

Proposition 52 *Tout automate de sable \mathfrak{F} -surjectif est surjectif.*

Preuve. On reprend la preuve de la proposition 51, en remplaçant x^n par $x^n \in \mathfrak{F}$ tel que pour tout $i \in \mathbb{Z}^d$, $|i| \leq n \Rightarrow x_i^n = x_i$ et $|i| > n \Rightarrow x_i^n = 0$. Le reste est inchangé. \square

La proposition suivante montre que la réciproque de la proposition 52 est fausse.

Proposition 53 *Il existe un automate de sable \mathfrak{P} -surjectif (donc surjectif grâce à la proposition 51) non \mathfrak{F} -surjectif.*

Preuve. Soit l'automate $\mathcal{L} = \langle 1, 1, f_{\mathcal{L}} \rangle$ avec :

$$\forall a, b \in \widetilde{[-1, 1]}, \quad f_{\mathcal{L}}(a, b) = \begin{cases} -1 & \text{si } a < 0, \\ +1 & \text{si } a > 0, \\ 0 & \text{sinon.} \end{cases}$$

La figure 26 illustre le fonctionnement de base de \mathcal{L} : chaque colonne va en direction de son voisin de gauche. Notons $F_{\mathcal{L}}$ sa règle globale.

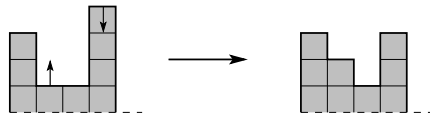


FIG. 26 – Fonctionnement de base de \mathcal{L} .

Montrons que \mathcal{L} n'est pas \mathfrak{F} -surjectif. Soit la configuration $x \in \mathfrak{F}$ définie par $x_0 = 2$ et $x_i = 0$ pour tout $i \neq 0$, et supposons qu'il existe une préimage $y \in \mathfrak{F}$ de x . Soit i le plus grand entier tel que $y_i \neq 0$. Comme $y_i \neq 0$ et $y_{i+1} = 0$, $x_{i+1} = F_{\mathcal{L}}(y)_{i+1} \neq 0$. Par conséquent $i = 0$ puisque x_0 est la seule valeur non nulle de x , mais alors $y_0 = 0$ et comme $f_{\mathcal{L}}$ ne peut renvoyer plus de 1, $x_0 = 2$ est inaccessible.

Il reste à prouver que \mathcal{L} est \mathfrak{P} -surjectif. Soit une configuration $x \in \mathfrak{P}$ de période $p \in \mathbb{N}^*$, construisons une de ses préimages. Il existe une unique suite croissante d'indices $(i_n)_{n \in [0, k]}$ tels que pour tout $n \in [0, k-1]$, $0 \leq i_n \leq i_{n+1} < p$, pour tout $n \in [0, k]$, $x_{i_n} \neq x_{i_{n+1}}$, et pour tout $i \in [i_n, i_{n+1} - 1]$, $x_i = x_{i_n}$. En d'autres termes, les i_n sont les indices qui correspondent aux variations de x dans l'intervalle $[0, p-1]$. L'idée est de travailler dans les intervalles délimités par ces indices en amplifiant la différence existant à la frontière, ce qui sera corrigé ensuite en appliquant la règle locale. Formellement, si $k < 0$ alors x est constante et est sa propre préimage périodique. Sinon pour tout $i \in [i_0, p + i_0 - 1]$, soit $n \in [0, k]$ tel que $i_n \leq i < i_{n+1}$ (on définit $i_{k+1} = i_0 + p$), et supposons que $x_{i_{n-1}} < x_{i_n}$ (si ce n'est pas le cas, on fait les opérations symétriques). On pose $y_i = x_i + 1$ si $i - i_n$ est pair, $y_i = x_i - 1$ si $i - i_n$ est impair. Cette opération est répétée exactement de la même manière sur les autres périodes de x , y est donc périodique de période p .

Montrons que $F_{\mathcal{L}}(y) = x$. Soit $i \in [i_0, p + i_0 - 1]$, supposons qu'il existe n tel que $i = i_n$. Alors $F_{\mathcal{L}}(y)_i = y_i + f_{\mathcal{L}}(R_1^i(y))$. Si l'on suppose que $x_{i-1} < x_i$ (encore une fois l'autre cas est symétrique), on a $y_i = x_i + 1 > x_{i-1} + 1$ d'où $y_i > y_{i-1}$ puisque $|x_{i-1} - y_{i-1}| \leq 1$. Par conséquent $f_{\mathcal{L}}(R_1^i(y)) = -1$ et $F_{\mathcal{L}}(y)_i = x_i + 1 - 1 = x_i$. Sinon, dans le cas où $i \neq i_n$ pour tout $n \in [0, k]$, on a par construction :

- (i) soit $y_i = x_i + 1$ et $y_{i-1} = x_{i-1} - 1$, et comme $x_i = x_{i-1}$ on obtient $y_i = y_{i-1} + 2$ et $F_{\mathcal{L}}(y)_i = x_i + 1 - 1 = x_i$;
- (ii) soit le cas symétrique $y_i = x_i - 1$ et $y_{i-1} = x_{i-1} + 1$ pour lequel on obtient de la même façon $F_{\mathcal{L}}(y)_i = x_i$.

Les configurations x et $F_{\mathcal{L}}(y)$ étant périodiques de même période p , elles sont aussi égales en dehors de l'intervalle $[i_0, p + i_0 - 1]$ donc \mathcal{L} est \mathfrak{P} -surjectif. \square

Les deux résultats suivants ne sont vrais qu'en dimension 1, et sont ouverts pour les dimensions supérieures.

Proposition 54 *Tout automate de sable surjectif est \mathfrak{P} -surjectif en dimension 1.*

Preuve. Soit \mathcal{A} un automate surjectif de rayon r , de fonction globale F , défini en dimension 1. Soit $x \in \mathfrak{P} \subset \mathfrak{E}$ une configuration périodique de période $p \in \mathbb{N}^*$. Soit $y \in \mathfrak{E}$ une préimage de x par F (F est surjective), construisons une configuration périodique z dont l'image est x . Soit $Y = \{(y_{\alpha p - r}, \dots, y_{\alpha p + r - 1}) \mid \alpha \in \mathbb{Z}\}$. Puisque pour tout $i \in \mathbb{Z}$, $|y_i - x_i| \leq r$ (la règle locale renvoie un entier entre $-r$ et r) et que x est périodique de période p , Y contient au plus $2r \cdot (2r + 1)$ éléments. Soit $k_1 = \alpha_1 p$ et $k_2 = \alpha_2 p$, $k_1 < k_2$, tels que $(y_{k_1 - r}, \dots, y_{k_1 + r - 1}) = (y_{k_2 - r}, \dots, y_{k_2 + r - 1})$. On définit la configuration $z \in \mathfrak{P}$ de période $k_2 - k_1$ par $z_{k_1 + i} = y_{k_1 + i}$ pour tout $i \in [0, k_2 - k_1 - 1]$ (voir figure 27).

En chaque point de z entre k_1 et $k_2 - 1$, le voisinage est identique à celui du même point dans y donc pour tout $j \in [k_1, k_2 - 1]$, $F(z)_j = F(y)_j = x_j$. Soit $i \in \mathbb{Z}$, il existe $j \in [k_1, k_2 - 1]$ et $\alpha \in \mathbb{Z}$ tels que $i = j + \alpha(k_2 - k_1)$. Par conséquent, $F(z)_i = F(z)_j = x_j = x_{j + p(\alpha(\alpha_2 - \alpha_1))} = x_i$: F est \mathfrak{P} -surjective. \square

Remarque 11 *Pour les dimensions supérieures à 1, la question de savoir si ce résultat est vrai ou pas est ouverte. Le problème est qu'au-delà de la dimension 1, le périmètre d'une boule (notre ensemble Y) contient un nombre d'éléments non borné, dépendant de la taille de la boule.*

Corollaire 55 *En dimension 1, tout automate de sable \mathfrak{F} -surjectif est \mathfrak{P} -surjectif.*

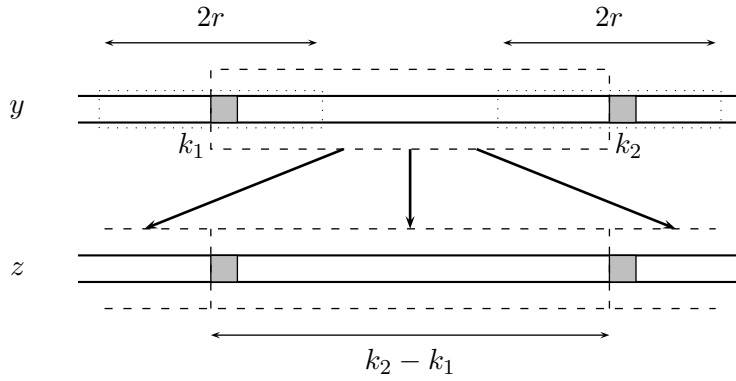


FIG. 27 – Construction de z à l'aide de y . Les voisinages de taille $2r$ autour de k_1 et k_2 dans y (en pointillés fins) contiennent les mêmes valeurs. On construit z en recopiant les éléments de y entre k_1 et k_2 (en pointillés longs) partout dans z .

Preuve. Un automate \mathfrak{F} -surjectif est surjectif (proposition 52) et donc \mathfrak{P} -surjectif en dimension 1 (corollaire 55). \square

Là encore ce problème est ouvert en dimension 2 et au-delà, et sa résolution semble être difficile.

Passons maintenant à l'étude de l'injectivité. Il est évident qu'un automate injectif est \mathfrak{F} -injectif et \mathfrak{P} -injectif. La réciproque est fausse, comme le montre la proposition suivante.

Proposition 56 *Il existe un automate de sable \mathfrak{F} -injectif, \mathfrak{P} -injectif mais pas injectif.*

Preuve. Soit l'automate de sable $\mathcal{Y} = \langle 1, 2, f_{\mathcal{Y}} \rangle$, avec :

$$\begin{aligned} \forall a, b, c \in \widetilde{[-2, 2]}, \quad & f_{\mathcal{Y}}(+\infty, a, b, c) = -1, \\ & f_{\mathcal{Y}}(2, a, b, c) = -1, \\ & f_{\mathcal{Y}}(1, a, b, c) = -1, \\ & f_{\mathcal{Y}}(0, a, b, c) = -1, \\ & f_{\mathcal{Y}}(-1, -\infty, a, b) = -1, \end{aligned}$$

et tout le reste renvoie 0. La règle globale de \mathcal{Y} est notée $F_{\mathcal{Y}}$, son comportement est difficile à identifier mais la figure 28 représente son fonctionnement sur deux configurations bien précises.

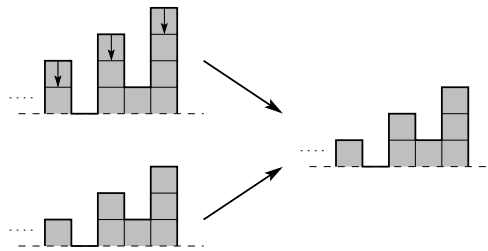


FIG. 28 – Fonctionnement de \mathcal{Y} sur deux configurations spécifiques.

Soient les deux configurations $x, y \in \mathfrak{C}$ définies par :

$$\forall i \in \mathbb{Z}, \quad \begin{cases} x_{2i} = i \\ x_{2i+1} = i + 3 \end{cases} \quad \text{et} \quad \begin{cases} y_{2i} = i \\ y_{2i+1} = i + 2 \end{cases},$$

elles sont partiellement dessinées sur la figure 28. Il est clair que $F_{\mathcal{Y}}(x) = F_{\mathcal{Y}}(y) = y$, \mathcal{Y} n'est donc pas injectif.

Pour montrer que \mathcal{Y} est \mathfrak{F} -injectif et \mathfrak{P} -injectif, on a besoin du résultat intermédiaire suivant : si $x, y \in \mathfrak{C}$ sont deux configurations distinctes telles que $F_{\mathcal{Y}}(x) = F_{\mathcal{Y}}(y)$, alors il y a une infinité de différences entre x et y , dont une infinité sont à des hauteurs différentes. En pratique nous montrons que si $x_i > y_i$, alors $x_{i-2} > y_{i-2}$ et $x_{i-2} < x_i$. Supposons pour cela qu'il existe deux configurations $x, y \in \mathfrak{C}$ telles que $x_i \neq y_i$ et $F_{\mathcal{Y}}(x) = F_{\mathcal{Y}}(y)$. On peut supposer sans perte de généralité que $x_i = y_i + 1$ ($f_{\mathcal{Y}}$ ne renvoie que 0 ou -1). Cela implique qu'une règle locale renvoyant 0 est appliquée à y à la position i , et donc $y_{i-2} \leq y_i - 1$ (puisque $f_{\mathcal{Y}}(a, -, -, -)$ renvoie 0 seulement si $a \leq -1$). De même, une règle renvoyant -1 est appliquée à x en i , d'où $x_{i-2} \geq y_i - 1$. On a donc

$$x_{i-2} \geq x_i - 1 = y_i \geq y_{i-2} + 1 > y_{i-2}. \quad (2.1)$$

La première conséquence de l'équation (2.1) est qu'il y a bien une infinité de différences entre x et y , il suffit de remonter de 2 en 2 à partir de la première différence trouvée. Comme deux configurations finies ne peuvent avoir qu'un nombre fini de différences, deux configurations finies distinctes ont nécessairement une image différente, $F_{\mathcal{Y}}$ est \mathfrak{F} -injective.

De plus, puisque $F_{\mathcal{Y}}(x) = F_{\mathcal{Y}}(y)$ et que $x_{i-2} > y_{i-2}$, on a $x_{i-2} = y_{i-2} + 1$. Les inégalités larges de l'équation (2.1) sont en réalité des égalités, en particulier $x_{i-2} = x_i - 1$. Par conséquent on obtient comme prévu $\dots < x_{i-4} < x_{i-2} < x_i$, qui suffit à prouver que deux configurations périodiques distinctes ont deux images distinctes (une configuration périodique contient un nombre fini de valeurs distinctes, ce qui serait contredit par l'inégalité précédente). \mathcal{Y} est donc également \mathfrak{P} -injectif. \square

Les deux propositions qui suivent permettent d'achever l'étude de l'injectivité.

Proposition 57 *Tout automate de sable \mathfrak{P} -injectif est \mathfrak{F} -injectif.*

Preuve. Prouvons cette proposition par contraposée. Soit un automate \mathcal{A} non \mathfrak{F} -injectif, de rayon r et de règle globale F . Soient $x^1, x^2 \in \mathfrak{F}$ deux configurations finies distinctes ayant la même image z par F . Soit $k \in \mathbb{N}$ tel que pour tout $i \in \mathbb{Z}^d$, $|i| > k \Rightarrow x_i^1 = x_i^2 = 0$. Construisons deux configurations périodiques distinctes en entourant les parties non nulles de x^1 et x^2 par une couronne de zéros d'épaisseur r , et en répétant ceci (voir figure 29 pour un exemple en dimension 2).

Pour $\alpha \in \{1, 2\}$, soit $y^\alpha \in \mathfrak{P}$ la configuration de période $(2k + 2r + 1, \dots, 2k + 2r + 1)$ définie par :

$$\forall i \in [-k - r, k + r]^d, \quad \begin{cases} y_i^\alpha = x_i^\alpha & \text{si } |i| \leq k, \\ y_i^\alpha = 0 & \text{si } k < |i| \leq k + r. \end{cases}$$

On remarque que $F(y^1) = F(y^2)$. Chaque point de la couronne de 0 se comporte de la même manière dans y^1 et y^2 , car son voisinage contient les mêmes valeurs que lorsqu'il est hors de la zone non nulle de x^1 et x^2 . Il en est de même pour les points à l'intérieur, qui se comportent exactement comme quand ils sont dans x^1 et x^2 . \mathcal{A} n'est donc pas non plus \mathfrak{P} -injectif. \square

Nous montrons maintenant que la réciproque de la proposition 57 est fausse, en complétant le résultat de la proposition 56.

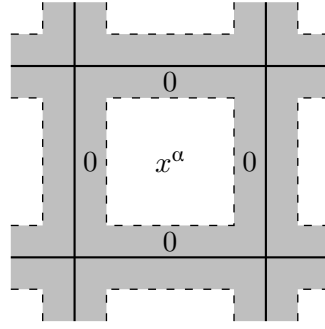


FIG. 29 – Construction de y^α en dimension 2, pour $\alpha \in \{1, 2\}$. Le blanc représente les valeurs non nulles de x^α , le gris les 0 ajoutés en périphérie.

Proposition 58 *Il existe un automate de sable qui est \mathfrak{F} -injectif mais ni injectif ni \mathfrak{P} -injectif.*

Preuve. Soit l'automate de sable $\mathcal{X} = \langle 1, 2, f_{\mathcal{X}} \rangle$ avec :

$$\begin{aligned} \forall a, b, c \in \widetilde{[-2, 2]}, \quad & f_{\mathcal{X}}(+\infty, a, b, c) = -1, \\ & f_{\mathcal{X}}(2, a, b, c) = -1, \\ & f_{\mathcal{X}}(1, -1, a, b) = -1, \\ & f_{\mathcal{X}}(1, -2, a, b) = -1, \\ & f_{\mathcal{X}}(1, -\infty, a, b) = -1, \\ & f_{\mathcal{X}}(0, -2, a, b) = -1, \\ & f_{\mathcal{X}}(0, -\infty, a, b) = -1, \end{aligned}$$

et toutes les autres valeurs possibles dans le voisinage renvoient 0. Notons $F_{\mathcal{X}}$ sa règle globale. Le comportement de cet automate est assez complexe en général, un exemple de son fonctionnement sur les deux configurations spécifiques qui nous intéressent est représenté sur la figure 30.

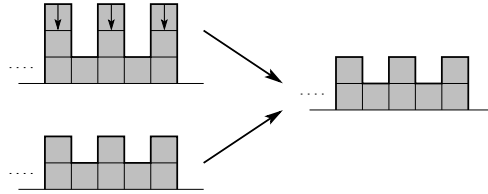


FIG. 30 – Exemple de fonctionnement de \mathcal{X} sur deux configurations spécifiques.

Montrons que \mathcal{X} n'est ni \mathfrak{P} -injectif ni injectif. Soient les deux configurations (périodiques) $x, y \in \mathfrak{P}$, $x \neq y$, définies comme suit (voir figure 30) :

$$\forall i \in \mathbb{Z}, \quad \begin{cases} x_{2i} = 0 \\ x_{2i+1} = 2 \end{cases} \quad \text{et} \quad \begin{cases} y_{2i} = 0 \\ y_{2i+1} = 1 \end{cases}.$$

On remarque que $F_{\mathcal{X}}(x) = F_{\mathcal{X}}(y) = y$, donc \mathcal{X} n'est pas \mathfrak{P} -injectif et bien sûr n'est pas non plus injectif.

Il reste à voir que \mathcal{X} est \mathfrak{F} -injectif, pour cela prenons deux configurations finies distinctes $x, y \in \mathfrak{F}$ et supposons que $F_{\mathcal{X}}(x) = F_{\mathcal{X}}(y)$. Étant donné que les configurations sont finies, on

peut choisir $i \in \mathbb{Z}$ comme étant le plus petit entier tel que $x_i \neq y_i$. Puisque $f_{\mathcal{X}}$ ne renvoie que 0 ou -1 , on sait que $|x_i - y_i| = 1$, et on peut supposer que $x_i = y_i + 1$ (l'autre cas est similaire). Ceci implique que la règle locale appliquée à x à la position i est l'une des sept qui renvoient la valeur -1 :

- si la portée est $(+\infty, -, -, -)$ (pour simplifier les notations, le symbole « $-$ » isolé indique un élément arbitraire de $[-2, 2]$), alors puisque $y_i = x_i - 1$ et $y_{i-2} = x_{i-2}$, la même règle est appliquée à y , ce qui signifie que $F_{\mathcal{X}}(x)_i \neq F_{\mathcal{X}}(y)_i$ qui est en contradiction avec l'hypothèse de départ ;
- si la portée est $(2, -, -, -)$, pour les mêmes raisons la règle correspondant à la portée $(+\infty, -, -, -)$ est appliquée à y , on obtient la même contradiction ;
- si le voisinage contient $(1, -1, -, -)$, $(1, -2, -, -)$ ou $(1, -\infty, -, -)$, la règle est appliquée à y avec la portée $(2, -, -, -)$, donc y_i est diminué de 1 et on a encore la même contradiction ;
- si la portée est $(0, -2, -, -)$ ou $(0, -\infty, -, -)$, comme $y_{i-2} = x_{i-2}$, $y_{i-1} = x_{i-1}$ et $y_i = x_i - 1$, la règle est appliquée à y en i avec l'une des portées $(1, -1, -, -)$, $(1, -2, -, -)$ ou $(1, -\infty, -, -)$, qui renvoient toutes -1 , on a encore une contradiction. \square

Les trois propositions qui suivent prouvent que contrairement à ce qu'il se passe pour les automates cellulaires [19], il n'y a pas de liens entre automates \mathcal{U} -surjectifs et automates \mathfrak{V} -surjectifs pour $\mathcal{U}, \mathfrak{V} \in \{\mathfrak{C}, \mathfrak{F}, \mathfrak{P}\}$.

Proposition 59 *L'automate de sable \mathcal{S} (page 52) est \mathcal{U} -surjectif pour $\mathcal{U} \in \{\mathfrak{C}, \mathfrak{F}, \mathfrak{P}\}$. L'automate \mathcal{S}^r (page 52) est \mathcal{U} -injectif pour $\mathcal{U} \in \{\mathfrak{C}, \mathfrak{F}, \mathfrak{P}\}$.*

Preuve. Pour les définitions de \mathcal{S} et \mathcal{S}^r , se reporter à la page 52. Comme $\mathcal{S} \circ \mathcal{S}^r = id$, \mathcal{S} est surjectif et \mathcal{S}^r est injectif. Notons que $\mathcal{S}^r \circ \mathcal{S} \neq id$, par exemple pour $x \in \mathfrak{C}$ telle que $x_0 = 2$ et $x_i = 0$ pour $i \neq 0$, $F_{\mathcal{S}^r}(F_{\mathcal{S}}(x)) = F_{\mathcal{S}}(x) \neq x$. De plus, comme \mathcal{S}^r construit une préimage pour \mathcal{S} , tout configuration finie [resp. périodique] a une préimage finie [resp. périodique]. L'injectivité de \mathcal{S}^r amène directement les derniers résultats. \square

Proposition 60 *L'automate \mathcal{S} n'est pas \mathcal{U} -injectif pour $\mathcal{U} \in \{\mathfrak{C}, \mathfrak{F}, \mathfrak{P}\}$.*

Preuve. Soient les configurations finies $x, y \in \mathfrak{F} \subset \mathfrak{C}$ telles que :

$$\forall i \in \mathbb{Z}, \quad x_i = 0 \quad \text{et} \quad y_i = \begin{cases} 1 & \text{si } i = 0, \\ -1 & \text{si } i = 1, \\ 0 & \text{sinon.} \end{cases}$$

Clairement $F_{\mathcal{S}}(x) = F_{\mathcal{S}}(y) = x$. De même soit $z \in \mathfrak{P}$ telle que $z_{2i} = 1$ et $z_{2i+1} = -1$ pour $i \in \mathbb{Z}$, on a $F_{\mathcal{S}}(x) = F_{\mathcal{S}}(z) = x$. \square

Proposition 61 *L'automate \mathcal{S}^r n'est pas \mathcal{U} -surjectif pour $\mathcal{U} \in \{\mathfrak{C}, \mathfrak{F}, \mathfrak{P}\}$.*

Preuve. Soit la configuration $x \in \mathfrak{F} \subset \mathfrak{C}$ telle que $x_0 = 2$ et $x_i = 0$ pour $i \in \mathbb{Z} \setminus \{0\}$. Supposons que x a une préimage y par $F_{\mathcal{S}^r}$. Le point y_0 peut contenir uniquement trois valeurs, puisque $f_{\mathcal{S}^r}$ renvoie $-1, 0$ ou $+1$.

- (i) Si $y_0 = 1$; alors $f_{\mathcal{S}^r}$ renvoie $+1$ donc $y_1 \leq y_0 - 2 = -1$. Alors $f_{\mathcal{S}^r}$ renvoie aussi $+1$ pour y_1 , donc $y_2 \leq -3$ ce qui n'est pas possible si l'on veut obtenir $x_2 = 0$.

(ii) Si $y_0 = 2$; alors la hauteur de la colonne est inchangée, ce qui signifie que ($y_{-1} \leq 3$ ou $y_1 \leq 0$) et ($y_{-1} \geq 4$ ou $y_1 \geq 1$). Comme pour le cas précédent, y_{-1} ne peut être plus grand que 4 pour atteindre $x_{-1} = 0$ donc $y_1 \geq 1$. La règle appliquée à y_1 renvoie -1 , d'où $y_0 \geq 3$, ce qui contredit l'hypothèse de départ.

(iii) Si $y_0 = 3$; alors f_{S^r} retourne -1 , d'où $y_{-1} \geq 5$, ce qui est encore une fois impossible.

S^r n'est donc ni surjectif ni \mathfrak{F} -surjectif. Pour montrer le résultat sur \mathfrak{P} , il suffit de prendre la configuration $x \in \mathfrak{P}$ de période 4 telle que $x_{4i} = 2$ pour tout $i \in \mathbb{Z}$, et $x_k = 0$ partout ailleurs. La preuve est semblable à celle sur \mathfrak{F} , la configuration se comportant exactement de la même manière sur l'intervalle $[-2, 2]$. \square

Les résultats obtenus dans cette section sont récapitulés par la figure 31.

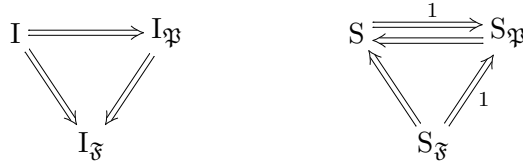


FIG. 31 – Relations entre les propriétés ensemblistes. I signifie injectivité et S signifie surjectivité. On note $I_{\mathfrak{U}}$ [resp. $S_{\mathfrak{U}}$] l'injectivité [resp. la surjectivité] restreinte aux configurations de \mathfrak{U} . Le symbole $\xRightarrow{1}$ indique une implication vraie en dimension 1, et un résultat inconnu pour les dimensions supérieures.

Pour les automates cellulaires, injectivité et surjectivité sont décidables en dimension 1 [1, 47]. La surjectivité et la réversibilité sont indécidables dans les dimensions supérieures ou égales à 2 [35, 18]. Nous pensons que la surjectivité et l'injectivité des automates de sable sont indécidables en dimension quelconque, mais nous n'avons pas encore trouvé de preuve. En effet comme on a pu le voir dans cette partie, il est difficile de se ramener à un problème plus simple et le problème général est très complexe.

2.2 Automates conservateurs de grains

Dans cette partie, nous étudions les automates de sable qui conservent les grains : pour ces automates, l'application de la règle globale sur une configuration finie ou périodique ne change pas le nombre de grains contenus dans la configuration. Cette classe est particulièrement intéressante, car elle contient tous les automates utiles du point de vue de la simulation des piles de sable : un modèle de piles de sable « déplace » les grains, sans les détruire et sans en créer.

Dans un premier temps, nous définissons cette notion intuitive de manière formelle. Pour cela nous montrons que les deux manières envisageables de compter les grains de sable sont équivalentes, ce qui définit parfaitement les automates conservateurs de grains. Ensuite, nous montrons que la conservation des grains est décidable (corollaire 66), en nous inspirant de la preuve faite pour les automates cellulaires dans [20], elle-même généralisant les résultats de [48, 5, 6]. Ce résultat se déduit d'une relation entre les différents éléments de la table de transition (théorème 65), relation qui n'existe que si l'automate de sable conserve les grains.

2.2.1 Définitions

Pour pouvoir « compter les grains » d'une configuration, il faut que celle-ci soit finie (nombre de grains finis) ou périodique (densité de grain constante). Une configuration finie restant finie après application de la règle globale d'un automate de sable, il est possible de comparer le nombre de grains avant et après. Une configuration périodique reste périodique et de même période, dans ce cas on peut comparer les densités, *i.e.* le nombre de grains présents sur une période. Ces deux notions sont formalisées par les deux définitions suivantes.

Définition 7 *Un automate de sable \mathcal{A} de règle globale F conserve les grains sur les finis si :*

$$\forall x \in \mathfrak{F}, \quad \sum_{i \in \mathbb{Z}^d} x_i = \sum_{i \in \mathbb{Z}^d} F(x)_i .$$

On dit alors que \mathcal{A} est FGC (finite grain conserving).

Définition 8 *Un automate de sable \mathcal{A} de règle globale F conserve les grains sur les périodiques si :*

$$\forall x \in \mathfrak{P}, \quad \sum_{\substack{i \in \mathbb{Z}^d \\ 0 \preccurlyeq i \prec p}} x_i = \sum_{\substack{i \in \mathbb{Z}^d \\ 0 \preccurlyeq i \prec p}} F(x)_i ,$$

où $p \in \mathbb{N}^{*d}$ est la période de x . \mathcal{A} est alors dit PGC (periodic grain conserving).

La proposition suivante montre que ces deux définitions sont équivalentes. Elle permet dorénavant de parler sans ambiguïté d'automates qui conservent les grains, de tels automates sont dits *GC* (grain conserving).

Proposition 62 *Les définitions FGC et PGC sont équivalentes.*

Preuve. Montrons que FGC implique PGC. Soit $\mathcal{A} = \langle d, r, f \rangle$ un automate FGC de rayon r , de règle globale F . Supposons que \mathcal{A} ne soit pas PGC, c'est-à-dire qu'il existe une configuration $x \in \mathfrak{P}$ de période $p = (p_1, \dots, p_d) \in \mathbb{N}^{*d}$ telle que $\sum_{0 \preccurlyeq i \prec p} x_i = k \neq k' = \sum_{0 \preccurlyeq i \prec p} F(x)_i$. On construit une configuration finie dérivée de x dans laquelle le nombre de grains n'est pas conservé. Soit x^p la matrice contenant tous les x_i tels que $i \in \mathbb{Z}^d$, $0 \preccurlyeq i \prec p$. Soit $y^\alpha \in \mathfrak{F}$ la configuration finie contenant α fois x^p dans chaque dimension, autour de quoi on place encore les r valeurs de x^p qui auraient dû se trouver là si on avait recopié encore une fois x^p . Partout ailleurs, y^α est remplie de 0. On a donc $y_i^\alpha = x_i$ pour tout $i \in \mathbb{Z}^d$ tel que $0 \preccurlyeq i \prec p + (2r, \dots, 2r)$ (voir figure 32 pour la dimension 2).

Soient $M = \max_{0 \preccurlyeq i \prec p} \max(x_i, F(x)_i)$ et $m = \min_{0 \preccurlyeq i \prec p} \min(x_i, F(x)_i)$ les deux valeurs extrêmes de x et de $F(x)$ à la fois. En comptant les grains de y^α , on obtient :

$$k\alpha^d + P(\alpha)m \leq \sum_{i \in \mathbb{Z}^d} y_i^\alpha \leq k\alpha^d + P(\alpha)M ,$$

où $P(\alpha) = \prod_{j=1}^d (\alpha p_j + 2r) - \prod_{j=1}^d (\alpha p_j)$ est le nombre d'éléments en périphérie (hachurés sur la figure 32). Notons que P est un polynôme de degré $d - 1$. De même, comptons les grains de $F(y^\alpha)$:

$$k'\alpha^d + Q(\alpha)m \leq \sum_{i \in \mathbb{Z}^d} F(y^\alpha)_i \leq k'\alpha^d + Q(\alpha)M ,$$

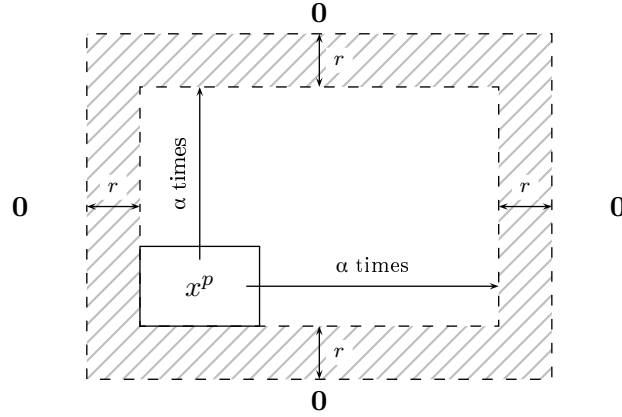


FIG. 32 – Construction d’une configuration finie y^α à partir de la configuration périodique x , en dimension 2.

où $Q(\alpha) = \prod_{j=1}^d (\alpha p_j + 4r) - \prod_{j=1}^d (\alpha p_j)$ est le nombre d’éléments non nuls de $F(y^\alpha)$ non centraux. Q est également un polynôme de degré $d - 1$.

Si $k > k'$, comme P et Q sont de degré $d - 1$, il est possible de trouver α suffisamment grand pour que :

$$\sum_{i \in \mathbb{Z}^d} y_i^\alpha \geq k\alpha^d + P(\alpha)m > k'\alpha^d + Q(\alpha)M \geq \sum_{i \in \mathbb{Z}^d} F(y^\alpha)_i .$$

De même, si $k < k'$, on peut trouver α tel que $\sum_{i \in \mathbb{Z}^d} y_i^\alpha < \sum_{i \in \mathbb{Z}^d} F(y^\alpha)_i$. Il y a contradiction avec le fait que \mathcal{A} est FGC, donc \mathcal{A} est PGC.

Réciproquement, soit $\mathcal{A} = \langle d, r, f \rangle$ un automate de sable PGC de règle globale F . Soient $x \in \mathfrak{F}$ et $k \in \mathbb{N}$ tel que pour tout $i \in \mathbb{Z}^d$, $|i| > k \Rightarrow x_i = 0$. Soit $y \in \mathfrak{P}$ la configuration de période $(2k + 2r + 1, \dots, 2k + 2r + 1)$ définie comme dans la preuve de la proposition 57 (voir figure 29, page 66) par :

$$\forall i \in [-k - r, k + r]^d, \quad \begin{cases} y_i = x_i & \text{si } |i| \leq k , \\ y_i = 0 & \text{si } k < |i| \leq k + r . \end{cases}$$

Par construction, pour tout point $i \in [-k - r, k + r]^d$ les cylindres de taille r centrés en i sont identiques dans x et dans y , donc $F(x)_i = F(y)_i$. On a alors :

$$\begin{aligned} \sum_{i \in \mathbb{Z}^d} F(x)_i &= \sum_{0 \leq |i| \leq k+r} F(x)_i = \sum_{0 \leq |i| \leq k+r} F(y)_i = \sum_{0 \leq |i| \leq k+r} y_i \quad (\mathcal{A} \text{ est PGC}) \\ &= \sum_{0 \leq |i| \leq k+r} x_i = \sum_{i \in \mathbb{Z}^d} x_i , \end{aligned}$$

\mathcal{A} est donc FGC. □

Remarque 12 Avec cette définition, les automates \mathcal{S} et \mathcal{S}^r (page 52) sont GC. C’est plutôt logique, car leur fonctionnement peut être décrit à l’aide de déplacements de grains (notamment pour \mathcal{S} qui simule SPM). Par contre, les automates vus dans la partie précédente (\mathcal{L} page 62, \mathcal{X} page 66, \mathcal{Y} page 64), ne le sont pas.

2.2.2 Décidabilité

Nous montrons dans cette partie que la conservation des grains est décidable pour les automates de sable, quels que soient leur dimension et leur rayon. Pour cela nous procédons comme cela a été fait pour les AC dans [20], en montrant le résultat sur les cas les plus simples et en les généralisant petit à petit.

Pour rendre les formules plus lisibles et pour faciliter les calculs, pour tout automate de sable $\mathcal{A} = \langle d, r, f \rangle$ nous introduisons une nouvelle fonction $g : \widetilde{\mathbb{Z}}^{[-r, r]^d} \mapsto [-r, r]$ définie par :

$$\forall x \in \mathfrak{C}, \forall i \in \mathbb{Z}^d, \quad g(N_r^i(x)) = f(R_r^i(x)) ,$$

où $N_r^i(x)$ est le *voisinage* de x centré en i de rayon r , il est défini tout simplement par $N_r^i(x)_j = x_{i+j}$ pour tout $j \in [-r, r]^d$. Par exemple, si l'on prend des voisinages de rayon 1 en dimension 1, $g(2, 3, 3) = g(-5, -4, -4) = f(-1, \perp, 0)$.

Commençons par énoncer et prouver la relation existant entre les différentes valeurs de la règle locale pour un automate de rayon 1 en dimension 1.

Proposition 63 *Un automate de sable $\mathcal{A} = \langle 1, 1, f \rangle$ est GC si et seulement si, pour tous $a, b, c \in \mathbb{Z}$:*

$$g(a, b, c) = g(0, 0, b) - g(0, 0, a) + g(0, b, c) - (0, a, b) .$$

Preuve. Soit $\mathcal{A} = \langle 1, 1, f \rangle$ un automate de sable de règle globale F . Soient $a, b, c \in \mathbb{Z}$, et $x = (\dots, 0, a, b, c, 0, \dots) \in \mathfrak{F}$. Comme \mathcal{A} est FGC (proposition 62), $\sum_{i \in \mathbb{Z}} F(x)_i = \sum_{i \in \mathbb{Z}} x_i = a + b + c$. Mais si l'on applique F à x on obtient également :

$$\sum_{i \in \mathbb{Z}} F(x)_i = g(0, 0, a) + g(0, a, b) + a + g(a, b, c) + b + g(b, c, 0) + c + g(c, 0, 0) ,$$

et donc :

$$g(a, b, c) = -g(0, 0, a) - g(0, a, b) - g(b, c, 0) - g(c, 0, 0) . \quad (2.2)$$

Afin de se débarrasser des termes $g(b, c, 0)$ et $g(c, 0, 0)$, on répète l'opération précédente sur la configuration $y = (\dots, 0, b, c, 0, \dots)$ (x privé de son premier élément). On a alors :

$$g(b, c, 0) = -g(0, 0, b) - g(0, b, c) - g(c, 0, 0) .$$

En injectant ce résultat dans l'équation (2.2) on obtient le résultat voulu.

Réciproquement, soit $\mathcal{A} = \langle 1, 1, f \rangle$ un automate de sable de règle globale F , vérifiant

$$\forall a, b, c \in \mathbb{Z}, \quad g(a, b, c) = g(0, 0, b) - g(0, 0, a) + g(0, b, c) - (0, a, b) . \quad (2.3)$$

Soit $x = (\dots, x_p, x_1, x_2, \dots, x_p, x_1, \dots) \in \mathfrak{P}$ une configuration périodique de période $p \in \mathbb{N}^*$. En calculant l'image de x par F , on a :

$$\sum_{i=1}^p F(x)_i = x_1 + g(x_p, x_1, x_2) + x_2 + g(x_1, x_2, x_3) + \dots + x_p + g(x_{p-1}, x_p, x_1) .$$

En remplaçant dans cette dernière équation les valeurs de g données par l'équation (2.3), tous les termes se simplifient et il reste $\sum_{i=1}^p F(x)_i = x_1 + \dots + x_p$, \mathcal{A} est donc GC. \square

Nous donnons maintenant une ébauche de preuve de la condition nécessaire et suffisante pour qu'un automate de dimension 2 soit GC. Les calculs étant lourds et peu intéressants, ils ne seront pas détaillés ici.

Proposition 64 *Un automate de sable $\mathcal{A} = \langle 2, r, f \rangle$ est GC si et seulement si, pour tous $(x_{i,j}) \in \mathbb{Z}^{[-r,r]^2}$:*

$$\begin{aligned}
 g \begin{pmatrix} x_{-r,-r} & \cdots & x_{r,-r} \\ \vdots & \ddots & \vdots \\ x_{-r,r} & \cdots & x_{r,r} \end{pmatrix} = & - \sum_{\substack{i=0 \\ i+j>0}}^{2r} \sum_{j=0}^{2r} g \begin{pmatrix} 0_{i,j} & & 0 \\ & x_{-r,-r} & \cdots & x_{r-i,-r} \\ 0 & \vdots & \ddots & \vdots \\ & x_{-r,r-j} & \cdots & x_{r-i,r-j} \end{pmatrix} \\
 & + \sum_{i=1}^{2r} \sum_{j=0}^{2r} g \begin{pmatrix} 0_{i,j} & & 0 \\ & x_{-r+1,-r} & \cdots & x_{r+1-i,-r} \\ 0 & \vdots & \ddots & \vdots \\ & x_{-r+1,r-j} & \cdots & x_{r+1-i,r-j} \end{pmatrix} \\
 & + \sum_{i=0}^{2r} \sum_{j=1}^{2r} g \begin{pmatrix} 0_{i,j} & & 0 \\ & x_{-r,-r+1} & \cdots & x_{r-i,-r+1} \\ 0 & \vdots & \ddots & \vdots \\ & x_{-r,r+1-j} & \cdots & x_{r-i,r+1-j} \end{pmatrix} \\
 & - \sum_{i=1}^{2r} \sum_{j=1}^{2r} g \begin{pmatrix} 0_{i,j} & & 0 \\ & x_{-r+1,-r+1} & \cdots & x_{r+1-i,-r+1} \\ 0 & \vdots & \ddots & \vdots \\ & x_{-r+1,r+1-j} & \cdots & x_{r+1-i,r+1-j} \end{pmatrix}
 \end{aligned}$$

où $0_{i,j}$ est la matrice à i colonnes et j lignes contenant 0 partout.

Ébauche de preuve. Soit un automate de sable \mathcal{A} de dimension 2, de règle globale F , qui conserve les grains. Soit $x \in \mathfrak{F}$ la configuration finie contenant la matrice

$$X = \begin{pmatrix} x_{-r,-r} & \cdots & x_{r,-r} \\ \vdots & \ddots & \vdots \\ x_{-r,r} & \cdots & x_{r,r} \end{pmatrix}$$

au centre et 0 partout ailleurs. En comptant les grains de $F(x)$ et de x et en les assimilant, on obtient comme dans la preuve de la proposition 63 une expression de $g(X)$ en fonction de différentes valeurs de g . Pour enlever les termes $g(Y)$ tels que $Y_{-r,-r} \neq 0$ (valeur en haut à gauche de Y non nulle), on répète ces opérations sur les configurations finies y et z , qui sont définies comme x sans la première ligne de x pour y , et sans la première colonne de x pour z . Enfin, on réitère cette opération sur la configuration finie t qui contient x sans la première ligne ni la première colonne. En injectant toutes ces nouvelles valeurs dans l'équation de départ on obtient le résultat voulu.

Réciproquement, si la formule est satisfaite il suffit pour chaque configuration périodique $x \in \mathfrak{P}$ de remplacer chaque terme du calcul de $F(x)$ par la somme donnée par la formule. Tous les termes s'annulent et il ne reste plus que les éléments x_i , $0 \preccurlyeq i \prec p$. \square

Comme on le voit, le fait d'ajouter une dimension complique énormément l'énoncé de notre condition. Afin de le simplifier dans le cas général, nous introduisons une nouvelle fonction

$\phi : \{0, 1\}^d \mapsto \mathbb{Z}$ définie par :

$$\forall a_1, \dots, a_d \in \{0, 1\}, \quad \phi(a_1, \dots, a_d) = \sum_{\substack{k_1=a_1 \\ k_1+\dots+k_d>0}}^{2r} \cdots \sum_{k_d=a_d}^{2r} g(M(k_1, \dots, k_d)) ,$$

où $M(k_1, \dots, k_d)$ est la matrice de dimension d à $(2r+1)^d$ éléments qui vérifie :

$$\forall i_1, \dots, i_d \in [0, 2r], \quad M(k_1, \dots, k_d)_{i_1, \dots, i_d} = \begin{cases} 0 & \text{si } i_1 \leq k_1 \text{ ou } \dots \text{ ou } i_d \leq k_d , \\ x_{-r+a_1+i_1-k_1-1, \dots, -r+a_d+i_d-k_d-1} & \text{sinon.} \end{cases}$$

En d'autres termes, M est remplie de 0 tant que tous les k_i ne sont pas atteints, puis commence avec les valeurs $x_{\dots, -r+a_i, \dots}$. Par exemple en dimension 1, on a :

$$\forall a \in \{0, 1\}, \quad \phi(a) = \sum_{k=1}^{2r} g(\underbrace{0, \dots, 0}_{k \text{ fois}}, x_{-r+a}, \dots, x_{r+a-k}) .$$

En dimension 2, ϕ se retrouve dans toutes les doubles sommes de l'énoncé de la proposition 64. Nous pouvons maintenant énoncer le théorème principal de manière plus simple.

Théorème 65 *Un automate de sable $\mathcal{A} = \langle d, r, f \rangle$ est GC si et seulement si, pour toutes les matrices $X = (x_{i_1, \dots, i_d}) \in \mathbb{Z}^{[-r, r]^d}$:*

$$g(X) = \sum_{a_1, \dots, a_d=0}^1 (-1)^{a_1+\dots+a_d+1} \phi(a_1, \dots, a_d) .$$

Ébauche de preuve. Il est illusoire d'envisager de faire une preuve rigoureuse de ce théorème, tant les calculs sont lourds. Néanmoins, l'idée générale de la preuve accompagnée des preuves des propositions 63 et 64 devrait suffire à convaincre du bon fonctionnement de celle-ci.

Étant donné un automate de sable GC de règle globale F et une matrice $X = (x_{i_1, \dots, i_d}) \in \mathbb{Z}^{[-r, r]^d}$, il faut encore une fois compter les grains avant et après application de F aux configurations finies contenant au centre les matrices $M(a_1, \dots, a_d)$ (voir ci-dessus les définitions de ϕ et de M) pour $a_1, \dots, a_d \in \{0, 1\}$, et des 0 partout ailleurs. Mettre une valeur a_i à 1 revient à enlever la première ligne de la i^e dimension de x . Comme $X = M(0, \dots, 0)$, on retrouve l'expression voulue de $g(X)$ après simplification des termes.

Pour la réciproque, il suffit de partir d'une configuration périodique et de compter les grains sur une période après application de la règle globale de l'automate. Si l'on remplace chaque terme en g par son équivalent avec des ϕ , tous les termes en ϕ se simplifient et il ne reste plus que le nombre de grains de départ. \square

Corollaire 66 *Pour tout automate de sable \mathcal{A} , il est possible de décider en temps fini si \mathcal{A} est GC.*

Preuve. Pour vérifier les conditions du théorème 65, il suffit de les vérifier pour un nombre fini de valeurs. En effet, pour construire toutes les portées possibles de rayon r , il suffit de fixer arbitrairement l'élément central x_0 à 0, puis de prendre les $2r+3$ valeurs possibles de $\widetilde{[-r, r]}$ pour

les autres éléments. Il faut également s'assurer que les différences entre deux éléments puissent prendre toutes les valeurs possibles de $[-r, r]$, pour les termes de droite de l'égalité.

Il suffit alors de vérifier un nombre exponentiel (mais fini) de valeurs : prenons le premier élément dans l'intervalle $[-r-1, r+1]$, le second dans $[-2r-2, 2r+2]$ et ainsi de suite jusqu'au $[(2r+1)^d - 1]^e$ élément. Il y a ainsi au maximum $\prod_{i=1}^{(2r+1)^d - 1} i(2r+3)$ tests à effectuer. \square

2.3 Automates ultimement périodiques et ultimement stables

Toujours dans le but d'identifier des automates simulant des phénomènes naturels de manière réaliste, nous essayons maintenant de caractériser les automates de sable ultimement périodiques et ultimement stables. Ces automates sont ceux qui après un nombre fini d'itérations envoient toutes les configurations sur une configuration périodique ou stable, *i.e.* l'un des comportements les plus simples du point de vue de la dynamique. Nous montrons qu'il est impossible de décider si un automate donné a un tel comportement [9, 10]. Pour poursuivre la comparaison avec les automates cellulaires, on sait que pour les AC ces propriétés sont également indécidables [21].

Étant donné un automate de sable \mathcal{A} de règle globale F , une configuration $x \in \mathfrak{C}$ est dite *ultimement périodique* pour F (ou pour \mathcal{A}) s'il existe $p \in \mathbb{N}^*$, $t \in \mathbb{N}$ tels que pour tous $i, k \in \mathbb{N}$, $F^{t+pk+i}(x) = F^{t+i}(x)$. On appelle alors t la *longueur du transitoire* de x , p sa *période* (temporelle). Si $p = 1$, x est dite *ultimement stable*. Un automate de sable \mathcal{A} est \mathfrak{U} -ultimement périodique [resp. stable] si pour tout $x \in \mathfrak{U}$, x est ultimement périodique [resp. stable] pour \mathcal{A} .

Ainsi, \mathcal{S} (page 52) est \mathfrak{F} -ultimement stable et \mathfrak{P} -ultimement stable. \mathcal{S}^r (page 52), \mathcal{X} (page 66) et \mathcal{Y} (page 64) ne sont ultimement périodiques sur aucun sous-ensemble non trivial de \mathfrak{C} , alors que \mathcal{L} (page 62) est \mathfrak{P} -ultimement périodique mais ni \mathfrak{F} -ultimement périodique ni ultimement stable sur aucun sous-ensemble non trivial de \mathfrak{C} .

Problème ULT-P(\mathfrak{U}) :

INSTANCE : un automate de sable $\mathcal{A} = \langle d, r, f \rangle$;

QUESTION : toutes les configurations de \mathfrak{U} sont-elles ultimement périodiques pour \mathcal{A} ?

Problème ULT-S(\mathfrak{U}) :

INSTANCE : un automate de sable $\mathcal{A} = \langle d, r, f \rangle$;

QUESTION : toutes les configurations de \mathfrak{U} sont-elles ultimement stables pour \mathcal{A} ?

Nous montrons en fin de partie que les quatre problèmes ULT-P(\mathfrak{F}), ULT-P(\mathfrak{P}), ULT-S(\mathfrak{F}) et ULT-S(\mathfrak{P}) sont indécidables (théorèmes 71 et 72). Pour cela nous réduisons ULT-P au problème de l'arrêt d'une machine à deux compteurs, initialisée avec les deux compteurs à 0 (la même réduction fonctionne de la même manière avec ULT-S). Les sections suivantes explicitent le fonctionnement de l'automate de sable simulant la machine à compteurs.

2.3.1 Construction de l'automate de sable

Définition 9 Une machine à deux compteurs est un quadruplet $\mathcal{M} = \langle Q, q_0, q_f, \delta \rangle$, où Q est un ensemble fini d'états, $q_0 \in Q$ est l'état initial, $q_f \in Q$ est l'état final. La fonction $\delta : Q \times \{0, 1\} \times \{0, 1\} \mapsto Q \times \{1, 2\} \times \{-1, 0, +1\}$ est la fonction de transition.

Dans notre preuve, on suppose que les deux compteurs R_1 et R_2 sont initialisés à 0. Si le deuxième argument de δ vaut 0, cela indique que R_1 contient 0. S'il vaut 1, c'est que R_1 n'est pas vide. De même, le troisième argument de δ indique si le compteur R_2 est à 0. δ renvoie le nouvel état, le numéro du compteur à modifier, et la variation à effectuer (diminution de 1, pas de changement, augmentation de 1). Par souci de clarté, les transitions $\delta(q, b_1, b_2) = (q', i, j)$ sont notées $(q', R_i + j)$.

Exemple 5 Soit la machine à deux compteurs \mathcal{M} définie par l'ensemble de règles suivant :

$$\begin{cases} \delta(q_0, 0, 0) &= (q_1, R_1 + 1) \\ \delta(q_1, 0, -) &= (q_f, R_1 + 0) \\ \delta(q_1, 1, -) &= (q_2, R_1 - 1) \\ \delta(q_2, -, -) &= (q_3, R_2 + 1) \\ \delta(q_3, -, -) &= (q_1, R_2 + 1) . \end{cases}$$

Les symboles « $-$ » représentent une valeur quelconque. \mathcal{M} commence par initialiser R_1 à 1, puis multiplie le contenu de R_1 par 2 et met le résultat dans R_2 avant d'atteindre l'état final.

On associe à chaque machine à deux compteurs \mathcal{M} dont les compteurs sont initialement à 0 un automate de sable $\mathcal{S}_{\mathcal{M}}$. Le principe de la construction de $\mathcal{S}_{\mathcal{M}}$ est illustré sur la figure 33.

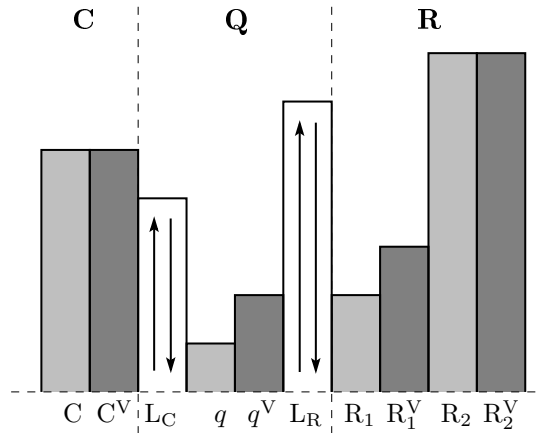


FIG. 33 – Simulation d'une machine à deux compteurs par un automate de sable.

L'idée générale est la suivante : $\mathcal{S}_{\mathcal{M}}$ utilise des piles de grains pour représenter les valeurs des compteurs (**R**) et de l'état de \mathcal{M} (**Q**). Pour des raisons techniques expliquées plus loin, on utilise un compteur supplémentaire (**C**) qui compte le nombre de pas effectués par \mathcal{M} depuis le début, et on duplique toutes les colonnes (celles avec les exposants V sur la figure 33).

Nous allons maintenant détailler les « astuces » qui assurent que la simulation se fait correctement.

Ascenseurs. La colonne q représentant l'état doit pouvoir envoyer des commandes aux compteurs C , R_1 et R_2 . Or le rayon de l'automate est fini, et ces compteurs peuvent avoir des valeurs arbitrairement grandes. On utilise donc des *ascenseurs* L_C et L_R (voir figure 33) qui vont monter pour transmettre les commandes à C et à R_1 et R_2 , puis redescendre.

Identité des colonnes. La règle locale de $\mathcal{S}_{\mathcal{M}}$ est constituée de plusieurs sous-règles, chacune d'entre elles concerne une colonne particulière de la simulation. Pour savoir quelle règle appliquer, il faut que chaque colonne de la configuration puisse savoir « qui elle est ». Ce n'est pas naturel dans les automates de sable, la règle locale étant invariante verticalement elle ne dépend pas de la hauteur de la colonne centrale. On décompose donc chaque colonne a en deux colonnes (a_l, a_r) telles que l'identité de a est codée par la différence (non nulle) entre a_l et a_r . Par exemple une différence de 1 pourrait signifier que a est le compteur C, 2 serait C^V et ainsi de suite. On utilise également un code pour un symbole d'erreur E, introduit dès que la configuration ne simule pas correctement une machine à compteurs (on y reviendra plus en détail dans la partie 2.3.3).

Par la suite, quand on parle de « hauteur » d'une colonne $a = (a_l, a_r)$, on fait référence à la hauteur de a_l . Les hauteurs (lors d'une simulation correcte) sont exprimées par rapport à la hauteur de la colonne q_l , colonne de gauche de l'état de la machine.

Commandes, couleurs et états. Puisque l'on peut coder de l'information dans la différence entre les deux composantes d'un élément de la simulation, en plus de l'identité on va coder des informations supplémentaires. Pour effectuer la simulation, on a besoin des commandes que les ascenseurs passent aux compteurs. Il faut notamment coder les commandes suivantes dans la différence entre les deux colonnes d'un ascenseur :

- C_{+1} qui permet d'augmenter C de 1 ;
- $C_{\rightarrow 0}^V$ pour réinitialiser C^V à 0 ;
- $R_{1,-1}$ diminue R_1 de 1 ;
- $L_{\rightarrow 0}$ indique à L_R ou L_C qu'il doit redescendre jusqu'à la colonne de référence.

On introduit également dans le codage des ascenseurs des couleurs : S, V_0, V, C pour indiquer où en est la simulation (commandes et couleurs sont détaillées dans la partie 2.3.2). Enfin, dans le codage de l'état on introduit également la valeur de l'état actuel.

Éviter les ambiguïtés. Soit N la valeur la plus importante permettant de coder de l'information (il y a un nombre borné de colonnes, de commandes, de couleurs, d'états, N est donc bien défini). Il faut s'assurer que si une différence entre deux colonnes consécutives est comprise entre 1 et N , alors elle code quelque chose. Par exemple la différence entre q_r et q_l^V doit toujours être strictement supérieure à N ou inférieure ou égale à 0 pour éviter qu'elles soient prises pour un codage qui n'a pas lieu d'être.

Pour cela, on utilise la technique suivante : dès que l'on augmente le contenu d'une colonne a de k grains (en référence à la machine à compteurs), on augmente en réalité a_l de $k \cdot (2N + 1)$ grains et a_r de $k \cdot (2N + 1) + \alpha$, $\alpha \in [-N + 1, N - 1]$ (α représente une éventuelle modification de l'information, par exemple un changement d'état). Cela crée des « lignes de niveau » (voir figure 34) sur toutes les hauteurs multiples de $2N + 1$. Toutes les colonnes de gauche se trouvent sur une de ces lignes de hauteur $k \cdot (2N + 1)$, celles de droite sont comprises entre les valeurs $k \cdot (2N + 1)$ et $k \cdot (2N + 1) + N$ pour un k donné. Cela garantit qu'entre une colonne de droite a_r et sa voisine de droite b_l , la différence n'est jamais comprise entre 1 et N (partie hachurée sur la figure 34). On est ainsi sûr que a_r est bien la colonne de droite et b_l celle de gauche, et ceci pour toutes les colonnes a et b de la simulation.

2.3.2 Simulation de la machine à deux compteurs

Chaque itération de \mathcal{M} est simulée par $\mathcal{S}_{\mathcal{M}}$ en trois phases.

S : *simulation* d'une itération de \mathcal{M} .

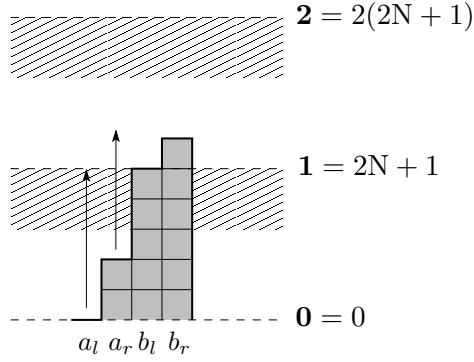


FIG. 34 – Lignes de niveau lors d’une simulation, avec $N = 2$. Les colonnes de gauche (indices l) se trouvent sur ces lignes en pointillés, alors que les colonnes de droite (indices r) peuvent se trouver n’importe où sauf dans la partie hachurée.

V : *vérification* depuis le début jusqu’à l’itération actuelle, dans les colonnes de vérification (exposant V).

C : *comparaison* des résultats obtenus par les deux premières étapes.

Remarque 13 *On utilise les mêmes notations pour les étapes de la simulation et pour les couleurs. Ce sont les couleurs des ascenseurs qui permettent de connaître l’étape courante de la simulation.*

Il faut maintenant expliciter en quoi consiste la phase de vérification. Il est en effet possible d’avoir pour l’automate $\mathcal{S}_{\mathcal{M}}$ une configuration codant une configuration de la machine \mathcal{M} telle que l’état soit valide et les valeurs des compteurs C , R_1 et R_2 soient cohérentes, sans pour autant que cette configuration soit atteignable. Par exemple, la configuration $(C, q = q_0, R_1 = 1, R_2 = 0)$ n’est pas atteignable par la machine de l’exemple 5 (page 75) à partir de q_0 avec les compteurs à 0. C’est pour cela qu’on ajoute à $\mathcal{S}_{\mathcal{M}}$ un compteur du nombre d’itérations et une phase de vérification, dans laquelle la simulation est effectuée du début jusqu’au nombre d’itérations voulues. Il suffit alors de vérifier que les contenus des colonnes avec ou sans V sont identiques, ce qui est fait pendant la phase de comparaison.

Les paragraphes suivants expliquent comment est effectuée chaque étape de la simulation.

Initialisation. Au début de la simulation d’un pas de la machine \mathcal{M} , la hauteur du compteur C représente le nombre de pas (de \mathcal{M}) effectués depuis le début. La colonne $q = (q_l, q_r)$ code l’état actuel de \mathcal{M} , les hauteurs des compteurs R_1 et R_2 sont positives et multiples de $2N + 1$. Les ascenseurs L_C et L_R sont à la hauteur 0 (relativement à q_l) et de couleur S . Les colonnes de vérification C^V , q^V , R_1^V et R_2^V peuvent contenir un nombre quelconque de grains.

S. Phase de simulation. Lors de cette étape, $\mathcal{S}_{\mathcal{M}}$ simule une seule itération de \mathcal{M} . Par exemple, supposons que l’on soit dans l’état q_1 , que R_1 et R_2 contiennent des valeurs strictement positives et que $\delta(q_1, 1, 1) = (q_2, R_i + j)$ pour $i \in \{1, 2\}$ et $j \in \{-1, 0, +1\}$. Alors $\mathcal{S}_{\mathcal{M}}$ doit changer q_1 en q_2 dans la colonne $q = (q_l, q_r)$, et en même temps faire partir L_C avec la commande C_{+1} et L_R avec la commande $R_{i,j}$. À titre d’exemple se trouvent ci-dessous les deux règles locales $f_{\mathcal{S}_{\mathcal{M}}}$

de $\mathcal{S}_{\mathcal{M}}$ qui effectuent la mise à jour des colonnes q_l et q_r correspondant à cette transition.

$$\begin{cases} f_{\mathcal{S}_{\mathcal{M}}}(\dots, \mathbf{L}_{\mathbf{C}}, \perp, \alpha_{q_1}, \underbrace{-, -}_{q^V}, \mathbf{L}_{\mathbf{R}}, \mathbf{R}_1, \underbrace{-, -}_{\mathbf{R}_1^V}, \mathbf{R}_2, \dots) = 0 & (\text{pour } q_l) , \\ f_{\mathcal{S}_{\mathcal{M}}}(\dots, \mathbf{L}'_{\mathbf{C}}, -\alpha_{q_1}, \perp, \underbrace{-, -}_{q^V}, \mathbf{L}'_{\mathbf{R}}, \mathbf{R}'_1, \underbrace{-, -}_{\mathbf{R}_1^V}, \mathbf{R}'_2, \dots) = \alpha_{q_2} - \alpha_{q_1} & (\text{pour } q_r) , \end{cases}$$

avec

$$\begin{aligned} \mathbf{L}_{\mathbf{C}} &= \underbrace{0, \alpha_{\mathbf{L}_{\mathbf{C}}, \mathbf{S}}}_{\mathbf{L}_{\mathbf{C}} \text{ en } 0, \text{ de couleur } \mathbf{S}} , & \mathbf{L}_{\mathbf{R}} &= \underbrace{0, \alpha_{\mathbf{L}_{\mathbf{R}}, \mathbf{S}}}_{\mathbf{L}_{\mathbf{R}} \text{ en } 0, \text{ de couleur } \mathbf{S}} , \\ \mathbf{L}'_{\mathbf{C}} &= \underbrace{-\alpha_{q_1}, \alpha_{\mathbf{L}_{\mathbf{C}}, \mathbf{S}} - \alpha_{q_1}}_{\mathbf{L}'_{\mathbf{C}} \text{ en } 0, \text{ de couleur } \mathbf{S}} , & \mathbf{L}'_{\mathbf{R}} &= \underbrace{-\alpha_{q_1}, \alpha_{\mathbf{L}_{\mathbf{R}}, \mathbf{S}} - \alpha_{q_1}}_{\mathbf{L}'_{\mathbf{R}} \text{ en } 0, \text{ de couleur } \mathbf{S}} , \end{aligned}$$

et

$$\begin{aligned} \mathbf{R}_1 &= \underbrace{> 0, > \alpha_{\mathbf{R}_1}}_{\mathbf{R}_1 \neq 0} , & \mathbf{R}_2 &= \underbrace{> 0, > \alpha_{\mathbf{R}_2}}_{\mathbf{R}_2 \neq 0} , \\ \mathbf{R}'_1 &= \underbrace{> -\alpha_{q_1}, > (\alpha_{\mathbf{R}_1} - \alpha_{q_1})}_{\mathbf{R}'_1 \neq 0} , & \mathbf{R}'_2 &= \underbrace{> -\alpha_{q_1}, > (\alpha_{\mathbf{R}_2} - \alpha_{q_1})}_{\mathbf{R}'_2 \neq 0} , \end{aligned}$$

où $1 \leq \alpha_a \leq N$ représente la différence qui code toutes les caractéristiques de la colonne a (identité, état, commande, couleur). La notation « $> x$ » signifie n'importe quel nombre strictement plus grand que x , et « $-$ » représente un nombre quelconque.

Les expressions étant vraiment complexes, on préférera désormais les décrire plutôt que de donner leur valeur précise. L'exemple ci-dessus montre qu'il n'est pas très difficile de « traduire » ces descriptions.

Dans la suite de la phase de simulation, les itérations de l'automate permettent aux ascenseurs de monter pour transmettre leur commande. Cela fait augmenter C de 1, et modifie également la valeur de l'un des compteurs R_1 ou R_2 si nécessaire. Ensuite, L_C et L_R redescendent grâce à la commande $L_{\rightarrow 0}$, tout en changeant leur couleur en V_0 . L'étape de simulation s'achève lorsque les deux ascenseurs ont rejoint la hauteur de référence et sont de couleur V_0 .

V_0 . Initialisation de la phase de vérification. Avant de démarrer la vérification, il faut réinitialiser toutes les colonnes de vérification à 0 (on rappelle qu'il s'agit de toutes les colonnes avec l'exposant V , en gris foncé sur la figure 33). On commence par fixer q^V à q_0 , en même temps que l'on passe les commandes $C_{\rightarrow 0}^V$ à L_C et $R_{\rightarrow 0}^V$ à L_R . La commande $R_{\rightarrow 0}^V$ déclenche une réaction en chaîne : L_R monte jusqu'à être au-dessus des deux compteurs, puis redescend en forçant les compteurs à l'accompagner. $C_{\rightarrow 0}^V$ a exactement les mêmes effets sur L_C et C .

Pour finir, quand les ascenseurs atteignent la hauteur de référence, ils prennent la couleur V pour indiquer que l'initialisation est terminée.

V. Phase de vérification. À chaque fois que les deux ascenseurs se trouvent au niveau de référence avec la couleur V , C itérations de \mathcal{M} sont effectuées dans les colonnes de vérification, en partant des compteurs R_1^V et R_2^V à 0. Cela se passe de la même manière que pour la phase S : les ascenseurs L_C et L_R transmettent les commandes aux compteurs C^V , R_1^V et R_2^V , pendant que l'état q^V évolue selon la fonction de transition de \mathcal{M} .

De plus, il faut que L_C détecte le moment où $C = C^V$, qui met fin à cette phase. Lorsque c'est le cas, il redescend avec la couleur C . Quand il atteint la hauteur de référence, L_R prend aussi la couleur C . On a alors $C = C^V$, et le triplet (q^V, R_1^V, R_2^V) devrait être égal au triplet (q, R_1, R_2) si la vérification a réussi. Les ascenseurs L_C et L_R sont à hauteur 0, de couleur C .

C. Phase de comparaison. L'ascenseur L_C est envoyé vers le haut jusqu'à ce que toutes les colonnes C , C^V , R_1 , R_1^V , R_2 , R_2^V soient plus basses que lui. Il commence alors à descendre en comparant les hauteurs des colonnes dès qu'il le peut.

Si L_C découvre une différence, il passe dans l'état d'erreur E , qui ne fait plus rien. La simulation est bloquée puisque toutes les autres colonnes attendent que l'ascenseur L_C redescende pour continuer d'évoluer, \mathcal{S}_M a atteint une configuration stable (donc temporellement périodique).

Si tout est correct, *i.e.* L_C atteint 0, L_C puis L_R prennent la couleur S et la simulation continue avec l'itération suivante de \mathcal{M} .

Complément sur la simulation. L'automate \mathcal{S}_M est bien défini de manière locale. Il suffit en effet d'un rayon $r = \max(3N + 1, 19)$ pour définir la règle locale. La valeur $3N + 1 = (2N + 1) + N$ permet à chaque couple de colonnes de calculer la différence de hauteur avec ses voisines lorsqu'elles sont à au plus une ligne de niveau de différence. La valeur 19 représente le nombre de colonnes moins une, cela permet à chaque colonne d'avoir la simulation toute entière dans son voisinage.

Enfin, pour toutes les portées de rayon r qui n'ont pas été décrites dans cette partie, la règle locale de \mathcal{S}_M renvoie 0 (à quelques exceptions près qui sont expliquées par la suite). Cette hypothèse est essentielle pour les preuves qui suivent.

2.3.3 Arrêt sur erreurs

Une configuration de \mathcal{S}_M est *valide* si elle contient une simulation correcte de \mathcal{M} (avec les compteurs initialisés à 0). Inversement, une configuration non valide est dite *mal formée*. Le but de cette partie est de montrer que toute configuration mal formée cesse d'évoluer au bout d'un temps fini. Ainsi, seules les configurations valides peuvent avoir un comportement qui n'est pas ultimement périodique.

Il y a deux sortes d'erreurs qui peuvent rendre une configuration mal formée. Il y a tout d'abord les *erreurs de voisinage*, qui ne sont pas dues à la colonne elle-même mais à sa situation. Par exemple, un couple de colonnes pourrait coder un compteur mais avoir une valeur négative. Ou alors des colonnes pourraient être mal placées, cela serait le cas par exemple s'il y avait deux colonnes d'état trop proches. Lorsque deux colonnes codent le symbole d'erreur E , il s'agit d'une autre forme d'erreur de voisinage.

Il peut aussi arriver que deux colonnes ne codent aucun symbole connu, ou qu'il y ait une ambiguïté sur ce codage. On parle alors d'*erreur d'identité*. Nous expliquons dans les deux paragraphes suivants comment sont gérées ces erreurs.

Erreurs de voisinage. Ces erreurs sont facilement détectables. Quand un couple de colonnes (a_l, a_r) repère des valeurs incohérentes dans son voisinage pendant la simulation, elle change son identité et devient le symbole d'erreur. La règle locale doit renvoyer 0 pour a_l et α pour a_r , α étant la valeur qui permet ce changement d'identité en E .

Erreurs d'identité. Dans tous les cas d'erreur d'identité, la règle locale retourne 0 à la fois pour la colonne de gauche et pour celle de droite. Cela concerne les colonnes dont aucune des colonnes adjacentes ne code de l'information (voir figure 35(a)), c'est-à-dire celles pour lesquelles la règle locale est $f_{\mathcal{S}_M}(\dots, x, \perp, y, \dots) = 0$, avec $x \geq 0$ ou $x < -N$ et $y \leq 0$ ou $y > N$. L'autre possibilité est lorsqu'on a ambiguïté dans le codage (codage valide sur la gauche et sur la droite,

voir figure 35(b)), la règle locale étant alors $f_{\mathcal{S}_{\mathcal{M}}}(\dots, x, \perp, y, \dots) = 0$, avec $-N \leq x < 0$ et $0 < y \leq N$.

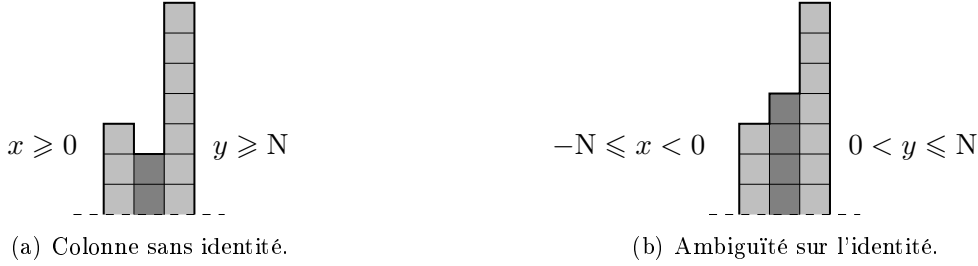


FIG. 35 – Types d'erreur d'identité.

Dorénavant, on note \mathcal{M} une machine à deux compteurs et $\mathcal{S}_{\mathcal{M}} = \langle 1, r, f_{\mathcal{S}_{\mathcal{M}}} \rangle$ l'automate de sable associé que nous venons de définir. Appelons $F_{\mathcal{S}_{\mathcal{M}}}$ la règle globale de $\mathcal{S}_{\mathcal{M}}$. Pour montrer qu'une configuration mal formée arrête d'évoluer au bout d'un temps fini, nous avons besoin des lemmes suivants.

Lemme 67 *Pour toute configuration $c \in \mathfrak{F}$, pour tout entier $t \in \mathbb{N}$, $|F_{\mathcal{S}_{\mathcal{M}}}^t(c)| \leq |c| + 1$.*

Preuve. Soit $c \in \mathfrak{F}$, soit $i = \max \{k \in \mathbb{Z} \mid c_k \neq 0\}$ le plus grand indice tel que c_i ne soit pas nul. Par construction, on a $f_{\mathcal{S}_{\mathcal{M}}}(\dots, -, \perp, 0, -, \dots) = 0$ donc pour tout $k \in \mathbb{Z}$, pour tout $t \in \mathbb{N}$, $k > i \Rightarrow F_{\mathcal{S}_{\mathcal{M}}}^t(c)_k = 0$.

Soit $j = \min \{k \in \mathbb{Z} \mid c_k \neq 0\}$. Remarquons que si la colonne c_{j-1} évolue, elle évolue nécessairement en tant que colonne de gauche d'un couple et reste donc sur les lignes de niveaux multiples de $2N + 1$. On a $f_{\mathcal{S}_{\mathcal{M}}}(\dots, -, 0, \perp, x, -, \dots) = 0$ pour $x \neq \alpha_C$, $0 < \alpha_C \leq N$ étant la différence de hauteur qui code la colonne $C = (C_l, C_r)$ (C_l est la seule colonne susceptible de voir une différence de 0 à gauche et d'évoluer quand même). Pour tout $k \in \mathbb{Z}$, pour tout $t \in \mathbb{N}$, il existe $n \in \mathbb{N}$ tel que $F_{\mathcal{S}_{\mathcal{M}}}^t(c)_{j-1} = n \cdot (2N + 1)$, donc $F_{\mathcal{S}_{\mathcal{M}}}^t(c)_{j-1} \neq \alpha_C$ et $k < j - 1 \Rightarrow F_{\mathcal{S}_{\mathcal{M}}}^t(c)_k = 0$. \square

Lemme 68 *Pour toute configuration $c \in \mathfrak{F}$, pour tout $i \in \mathbb{Z}$, si (c_i, c_{i+1}) codent une identité de colonne I alors pour tout $t \in \mathbb{N}$, les colonnes $(F_{\mathcal{S}_{\mathcal{M}}}^t(c)_i, F_{\mathcal{S}_{\mathcal{M}}}^t(c)_{i+1})$ codent la même identité I.*

Preuve. Soit $c \in \mathfrak{F}$ une configuration contenant un couple de colonnes d'identité I aux positions $(i, i + 1)$. Tout au long d'une simulation correcte, ce couple évolue selon les règles locales qui peuvent changer son état ou sa couleur, mais jamais son identité I.

Si la simulation n'est pas correcte, il pourrait y avoir un problème si une colonne voisine se rapproche trop de c_i ou c_{i+1} (figure 36). On se trouve avec une erreur d'identité due à l'ambiguïté. Il suffit alors de rectifier les règles locales pour qu'une colonne d'un couple ne se déplace que s'il n'y a pas d'erreur d'identité sur les deux colonnes du couple. Sur la figure 36, cela signifie qu'on empêche c_i de se déplacer car elle détecte une ambiguïté pour c_{i+1} . Cette contrainte n'affecte pas la simulation, car cela ne doit pas arriver pendant une simulation correcte. Dans tous les cas incorrects, les colonnes c_i et c_{i+1} ne sont donc jamais modifiées et I est préservé. \square

Lemme 69 *Pour toute configuration $c \in \mathfrak{F}$ qui n'est pas ultimement stable pour $\mathcal{S}_{\mathcal{M}}$, il existe dans c un ascenseur dont la couleur change infiniment souvent.*

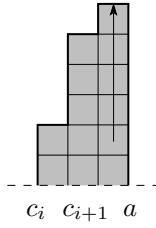


FIG. 36 – Conservation de l'identité : l'identité codée par (c_i, c_{i+1}) n'est pas modifiée par l'apparition de a dans le voisinage.

Preuve. Soit $c \in \mathfrak{F}$ non ultimement stable pour $\mathcal{S}_{\mathcal{M}}$. D'après le lemme 67, pour tout $t \in \mathbb{N}$, $|\mathcal{F}_{\mathcal{S}_{\mathcal{M}}}^t(c)|$ est borné indépendamment de t . Le comportement infini de c est donc dû à du mouvement « vertical », *i.e.* il y a une colonne c_i , $i \in \mathbb{Z}$, dont la valeur varie infiniment souvent. Cela implique qu'elle code une identité correcte, et d'après le lemme 68 celle-ci n'est pas modifiée. Il y a donc un ascenseur dont la valeur varie infiniment souvent. En effet les ascenseurs sont les « horloges » de la simulation, un couple de colonnes ne se déplace que s'il y a un ascenseur dans son voisinage qui le lui permet.

De plus, comme les règles ne permettent pas aux ascenseurs d'aller plus haut que la colonne la plus haute, ni plus bas que la colonne la plus basse, et qu'il n'y a pas de colonnes infinies dans c , l'ascenseur que l'on a mis en évidence passe infiniment souvent par la valeur du point de référence. D'après la définition des règles locales (description de la simulation dans la partie 2.3.2), il change de couleur à chaque passage par ce point. \square

Nous pouvons maintenant montrer que seules les configurations valides sont susceptibles de ne pas être ultimement périodiques.

Proposition 70 *Toute configuration $c \in \mathfrak{F}$ mal formée est ultimement stable pour $\mathcal{S}_{\mathcal{M}}$.*

Preuve. Soit $c \in \mathfrak{F}$ mal formée, on suppose que c n'est pas ultimement stable. D'après le lemme 69 il y a dans c un ascenseur qui change infiniment souvent de couleur. Il y a donc nécessairement autour de cet ascenseur une zone de c qui effectue la simulation de \mathcal{M} correctement, sinon l'ascenseur se serait arrêté. En effet lors de son évolution infinie il a plusieurs fois l'occasion de vérifier que toutes les colonnes ont des contenus valides, notamment dans la phase de comparaison C (qui revient infiniment souvent) où L_C vérifie successivement tous les couples de colonnes. \square

2.3.4 Décidabilité

Théorème 71 *Les problèmes $ULT-P(\mathfrak{F})$ et $ULT-P(\mathfrak{P})$ sont indécidables.*

Preuve. Remarquons tout d'abord qu'il suffit de prouver l'énoncé pour \mathfrak{F} , la preuve pour \mathfrak{P} est similaire : à partir d'une configuration finie on peut obtenir une configuration périodique en répétant la partie non nulle à intervalles réguliers. Il suffit également de montrer ce résultat en dimension 1, la même construction pouvant être effectuée dans les dimensions supérieures.

Nous réduisons ce problème au problème de l'arrêt d'une machine à deux compteurs initialisée à 0, problème que l'on sait indécidable. Soit \mathcal{M} une machine à compteurs, soit $\mathcal{S}_{\mathcal{M}}$ l'automate de sable qui simule \mathcal{M} selon la construction qui précède. Si \mathcal{M} ne s'arrête pas, alors il existe une configuration finie qui n'est pas ultimement périodique pour $\mathcal{S}_{\mathcal{M}}$: il suffit de prendre la

configuration qui code l'entrée $C = C^V = 0$, $q = q^V = q_0$, $R_1 = R_1^V = R_2 = R_2^V = 0$. À partir de là \mathcal{M} va être simulée correctement à l'infini, le compteur va augmenter indéfiniment, empêchant tout comportement périodique.

Réciproquement, supposons qu'il existe une configuration $c \in \mathfrak{F}$ qui n'est pas ultimement périodique pour $\mathcal{S}_{\mathcal{M}}$, donc pas ultimement stable. D'après la proposition 70, c est valide. De plus, à partir du lemme 69, on sait que le comportement infini a lieu sur une des parties valides de c . L'évolution de c contient donc au moins une simulation correcte, qui représente une évolution infinie de \mathcal{M} . En d'autres termes, \mathcal{M} ne s'arrête pas sur l'entrée 0. \square

On montre le résultat suivant avec exactement la même construction et la même preuve que pour le théorème 71.

Théorème 72 *Les problèmes $ULT-S(\mathfrak{F})$ et $ULT-S(\mathfrak{P})$ sont indécidables.*

Conclusions et perspectives

Dans ce travail de thèse, nous nous sommes intéressés à des systèmes dynamiques discrets particuliers : les modèles de piles de sable. Parmi eux, SPM et $\text{IPM}(k)$ (page 8) sont certainement les plus connus et les mieux maîtrisés. Pour ces modèles, on sait qu'en partant d'une configuration initiale où tous les grains se trouvent empilés verticalement sur une seule colonne, l'ensemble des configurations atteignables forme un treillis. Il est également possible de caractériser les éléments de ce treillis, ainsi que la longueur entre le point de départ et le point fixe. Ces résultats manquant de généralité, nous avons essayé d'étudier plusieurs extensions.

D'une part, nous avons vu une manière de généraliser leur étude à des piles de sables initiales quelconques, à l'aide d'un algorithme qui calcule rapidement leur point fixe (pseudo-code 1, théorème 15). Cet algorithme présente l'intérêt de pouvoir s'adapter à n'importe quel modèle de piles de sable, pourvu que la structure du graphe des orbites soit un treillis (unicité du point fixe). La vitesse de l'algorithme (proposition 17) dépend alors du temps nécessaire pour identifier les configurations atteignables à partir d'une unique colonne de grains.

D'autre part, nous avons introduit SSPM, un nouveau modèle de piles de sable symétriques. Il possède des propriétés particulièrement intéressantes : bien que le point fixe d'une pile où tous les grains sont concentrés dans une seule colonne ne soit pas unique, nous avons donné une caractérisation précise permettant de les construire (propositions 26 et 28) et de les compter (proposition 32). Nous complétons l'étude de SSPM par le calcul de la longueur des transitoires permettant d'atteindre les points fixes. Ces résultats font de notre modèle symétrique un modèle plus réaliste, prolongeant de manière naturelle SPM.

Enfin, nous terminons en abordant le problème de la généralisation des piles de sable d'un point de vue différent. En introduisant une topologie sur l'espace des configurations, on a pu définir les automates de sable. Ces nouveaux systèmes permettent de simuler tous les modèles de piles de sable définis localement. Les automates de sable sont les fonctions qui agissent de manière continue sur l'espace des configurations, qui commutent avec les décalages horizontaux et verticaux en préservant les infinis (théorème 47). Ce formalisme nous a permis une étude plus abstraite et donc plus générale des modèles de piles de sable. Nous avons vu qu'il est difficile de prévoir la dynamique de ces automates, même en se restreignant aux classes d'automates simulant des modèles « réalistes » : si l'on est capable de dire si un automate particulier conserve les grains (corollaire 66), il est impossible de savoir à l'avance s'il va se stabiliser ou atteindre un fonctionnement périodique (théorèmes 71 et 72).

Les solutions que nous avons apportées aux problèmes posés ouvrent de nouvelles voies pour les recherches futures. En ce qui concerne les piles de sable, le modèle SSPM mériterait d'être étendu de plusieurs manières. Pour refaire le lien avec les piles de sable généralisées, il faudrait pouvoir trouver un moyen d'accélérer la simulation de SSPM en partant d'une pile initiale quelconque. Nous avons vu que l'algorithme que nous proposons pour SPM ne suffit pas à donner tous les points fixes possibles, il faudrait donc aborder cette question différemment.

De plus, il serait intéressant d'étudier les extensions naturelles de SSPM, qu'il s'agisse de SIPM(k) ou de modèles multi-dimensionnels. Nous avons déjà donné un premier aperçu des propriétés que l'on peut espérer obtenir (notamment le nombre de points fixes). L'objectif final serait d'avoir des résultats concernant un modèle en dimension quelconque, où les grains pourraient tomber et glisser dans toutes les directions à la fois. Si l'on arrivait à trouver un modèle satisfaisant, on pourrait alors envisager d'étudier les variantes parallèles et déterministes de ce modèle. Pour l'instant, notre étude de SSPM (et plus généralement celle de SPM et IPM(k)) se limite à un fonctionnement séquentiel, non déterministe. On aurait alors un modèle dont le fonctionnement s'approcherait plus de ce qu'on s'imagine être la réalité, et qui pourrait être implémenté à l'aide d'automates de sable.

Il conviendrait également de tester le réalisme de tous ces systèmes dynamiques d'un point de vue physique. Peut-on retrouver avec nos modèles discrets les lois de la physique qui régissent les piles de sable « réelles » ? Dans la réalité, les éboulements des piles de sable sont modélisés à l'aide d'équations différentielles. La discrétisation permet d'approximer ces équations pour les rendre plus simples, mais il faudrait s'assurer que les erreurs d'approximation ne sont pas trop importantes.

Pour ce qui est des automates de sable, les perspectives sont encore plus nombreuses étant donné que leur introduction est très récente. Il faudrait compléter les résultats de cette thèse, notamment au sujet de la dynamique : est-ce que surjectivité et injectivité sont décidables ? Pour les automates cellulaires, les algorithmes de décision [1, 47] en dimension 1 sont basés sur le fait que le nombre d'états est borné. Les preuves d'indécidabilité dans les dimensions supérieures à 2 à l'aide de pavages [35] fonctionnent à condition que chaque point connaisse son propre état. Aucune de ces contraintes n'étant vérifiée pour les automates de sable, l'adaptation de ces résultats en dimension quelconque semble difficile.

Il serait également intéressant d'arriver à définir les automates de sable nilpotents, sous-classe des automates ultimement stables. Comme nous l'avons vu (page 60), nous avons des exemples d'automates de sable (\mathcal{L} par exemple) vérifiant cette notion intuitive, mais qui ne satisfont pas la définition classique à partir des ensembles limites [33, 16]. Une solution envisagée pour contourner ce problème est de considérer les μ -ensembles limites associés aux mesures probabilistes μ (voir [38]). Ces ensembles ne contiennent pas les configurations dont le nombre de préimages est trop faible, en d'autres termes on enlève les configurations « défectueuses ». Il ne reste alors plus que les configurations représentatives, celles dont le nombre de préimages est non négligeable relativement à la mesure probabiliste μ choisie.

En revenant à des aspects topologiques, on aimerait enfin examiner les classes de Kůrka [37] des automates cellulaires (équicontinuité, sensibilité aux conditions initiales, expansivité). Il faudrait commencer par adapter leur définition aux automates de sable et à leurs spécificités, mais même ainsi on n'est pas sûr de pouvoir les peupler toutes. On n'a en effet toujours pas d'exemple non trivial d'automate équicontinu, transitif ou expansif (par contre, \mathcal{S} et \mathcal{L} sont sensibles aux conditions initiales).

Nous pouvons pour terminer nous poser des questions plus éloignées de ce que nous avons fait dans ce travail de thèse. Par exemple, un problème majeur dans les automates cellulaires a été de trouver une classification homogène permettant de regrouper les automates ayant un comportement « semblable ». Nous avons vu qu'il existait une grande diversité de comportements possibles avec les automates de sable, il serait légitime d'essayer de les classer selon des critères dynamiques à déterminer. Ces classes devraient être définies de sorte que chacune soit représentative d'un type de comportement bien précis. Ensuite, en se restreignant aux automates d'une classe particulière, on peut espérer que les problèmes généraux se simplifient pour donner

de nouveaux résultats. Si cela ne suffit pas, on peut introduire de nouvelles contraintes, par exemple en se limitant aux automates conservateurs de grains d'une classe précise (voir partie II, section 2.2).

Enfin, une piste de recherche qui reste à explorer est l'aspect computationnel de ces automates. Nous savons déjà qu'ils ont la puissance de calcul des automates cellulaires (proposition 44) et donc des machines de Turing. Il reste à trouver les problèmes que leurs particularités (nombre d'états non borné, fonctionnement synchrone *etc.*) permettent de résoudre plus efficacement. Les calculs les plus simples sont les calculs de fonction caractéristique : à partir d'une entrée w sur un alphabet Σ , le modèle de calcul considéré doit atteindre en un temps fini un état d'acceptation ou de refus. Dans la littérature, on trouve plusieurs façons de définir le codage de l'entrée ainsi que le mode de reconnaissance. Pour les automates de sable, les solutions à retenir seraient vraisemblablement l'entrée séquentielle (mot inséré lettre par lettre, à chaque instant, sur la colonne centrale, voir [12]) ou parallèle (mot inséré entièrement dans la configuration initiale, voir [45, 11] par exemple). Pour l'acceptation, on pourrait convenir que la valeur -1 dans la colonne centrale indique que le mot est accepté, -2 qu'il est refusé. L'intérêt des automates de sable pour la reconnaissance de langages est évident : malgré quelques limitations dues au rayon fini de l'automate, on pourrait reconnaître des langages sur \mathbb{N} , un alphabet infini. À terme, on espère ainsi pouvoir mettre en évidence plusieurs classes de complexité de langages, selon leur mode et leur temps de reconnaissance par automate de sable (mode séquentiel ou parallèle ; temps réel, linéaire, polynômial ou exponentiel).

Bibliographie

- [1] S. AMOROSO : Decision procedures for surjectivity and injectivity of parallel maps for tessellation structures. *Journal of Computer and System Sciences*, 6(5):448–464, 1972.
- [2] G. E. ANDREWS : The theory of partitions. In *Encyclopedia of Mathematics and its Applications*, volume 2. Addison-Wesley, 1976.
- [3] P. BAK, C. TANG et K. WIESENFELD : Self-organized criticality. *Physical Review A*, 38(1):364–374, 1988.
- [4] G. BIRKHOFF : *Lattice theory*, volume 25 de *American Mathematical Society Colloquium Publications*. American Mathematical Society, 1967.
- [5] N. BOCCARA et H. FUKS : Cellular automaton rules conserving the number of active sites. *Journal of Physics A*, 31(28):6007–6018, 1998.
- [6] N. BOCCARA et H. FUKS : Number-conserving cellular automaton rules. *Fundamenta Informaticae*, 52:1–13, 2001.
- [7] T. BRYLAWSKI : The lattice of integer partitions. *Discrete Mathematics*, 6:201–219, 1973.
- [8] J. CERVELLE et E. FORMENTI : On sand automata. In *Symposium on Theoretical Aspects of Computer Science (STACS 2003)*, volume 2607 de *Lecture Notes in Computer Science*, pages 642–653. Springer-Verlag, 2003.
- [9] J. CERVELLE, E. FORMENTI et B. MASSON : Basic properties for sand automata. In *Mathematical Foundations of Computer Science (MFCS 2005)*, volume 3618 de *Lecture Notes in Computer Science*, pages 192–211. Springer-Verlag, 2005.
- [10] J. CERVELLE, E. FORMENTI et B. MASSON : From sandpiles to sand automata. Soumis, 2006.
- [11] C. CHOFRUT et K. CULIK II : On real-time cellular automata and trellis automata. *Acta Informatica*, 21(4):393–407, 1984.
- [12] S. N. COLE : Real-time computation by n -dimensional iterative arrays of finite-state machines. *IEEE Transactions on Computers*, C-18(4):349–165, 1969.
- [13] R. CORI et D. ROSSIN : On the sandpile group of a graph. *European Journal of Combinatorics*, 21(4):447–459, 2000.
- [14] R. CORI, D. ROSSIN et B. SALVY : Polynomial ideals for sandpiles and their Gröbner bases. *Theoretical Computer Science*, 276:1–15, 2002.
- [15] S. CORTEEL et D. GOUYOU-BEAUCHAMPS : Enumeration of sand piles. *Discrete Mathematics*, 256(3):625–643, 2002.
- [16] K. CULIK, J. PACHL et S. YU : On the limit sets of cellular automata. *SIAM Journal on Computing*, 18(4):831–842, 1989.

- [17] D. DHAR, P. RUELLE, S. SEN et D. VERMA : Algebraic aspects of Abelian sandpile models. *Journal of Physics A*, 28(4):805–831, 1995.
- [18] B. DURAND : The surjectivity problem for 2D cellular automata. *Journal of Computer and System Sciences*, 49(3):718–725, 1994.
- [19] B. DURAND : Global properties of cellular automata. *In Cellular Automata and Complex Systems*. Kluwer, 1998.
- [20] B. DURAND, E. FORMENTI et Z. RÓKA : Number conserving cellular automata I: decidability. *Theoretical Computer Science*, 299:523–535, 2003.
- [21] B. DURAND, E. FORMENTI et G. VAROUCHAS : On undecidability of equicontinuity classification for cellular automata. *In Discrete Models for Complex Systems (DMCS 2003)*, volume AB de *DMTCS Proceedings*, pages 117–128. Discrete Mathematics and Theoretical Computer Science, 2003.
- [22] J. O. DURAND-LOSE : Parallel transient time of one-dimensional sand pile. *Theoretical Computer Science*, 205:183–193, 1998.
- [23] E. FORMENTI et B. MASSON : A note on fixed points of generalized ice pile models. *International Journal of Unconventional Computing*, 2(2), 2006.
- [24] E. FORMENTI et B. MASSON : On computing fixed points for generalized sandpiles. *International Journal of Unconventional Computing*, 2(1):13–25, 2006.
- [25] E. FORMENTI, B. MASSON et T. PISOKAS : Advances in symmetric sandpiles. *Fundamenta Informaticae*, 76:1–22, 2006.
- [26] E. FORMENTI, B. MASSON et T. PISOKAS : On symmetric sandpiles. *In Cellular Automata for Research and Industry (ACRI 2006)*, volume 4173 de *Lecture Notes in Computer Science*, pages 676–685. Springer-Verlag, 2006.
- [27] E. GOLES et M. A. KIWI : Games on line graphs and sandpile automata. *Theoretical Computer Science*, 115:321–349, 1993.
- [28] E. GOLES, M. LATAPY, C. MAGNIEN, M. MORVAN et H. D. PHAN : Sandpile models and lattices: a comprehensive survey. *Theoretical Computer Science*, 322:383–407, 2004.
- [29] E. GOLES, M. MORVAN et H. D. PHAN : Sandpiles and order structure of integer partitions. *Discrete Applied Mathematics*, 117(1–3):51–64, 2002.
- [30] E. GOLES, M. MORVAN et H. D. PHAN : The structure of linear chip firing games and related models. *Theoretical Computer Science*, 270:827–841, 2002.
- [31] C. GREENE et D. J. KLEIMAN : Longest chains in the lattice of integer partitions ordered by majorization. *European Journal of Combinatorics*, 7(1):1–10, 1986.
- [32] G. A. HEDLUND : Endomorphisms and automorphisms of the shift dynamical system. *Mathematical Systems Theory*, 3(4):320–375, 1969.
- [33] L. P. HURD : Formal language characterization of cellular automaton limit sets. *Complex Systems*, 1(1):69–80, 1987.
- [34] J. KARI : The nilpotency problem of one-dimensional cellular automata. *SIAM Journal on Computing*, 21(3):571–586, 1992.
- [35] J. KARI : Reversibility and surjectivity problems of cellular automata. *Journal of Computer and System Sciences*, 48(1):149–182, 1994.
- [36] P. KÚRKA : Languages, equicontinuity and attractors in cellular automata. *Ergodic Theory & Dynamical Systems*, 17:417–433, 1997.

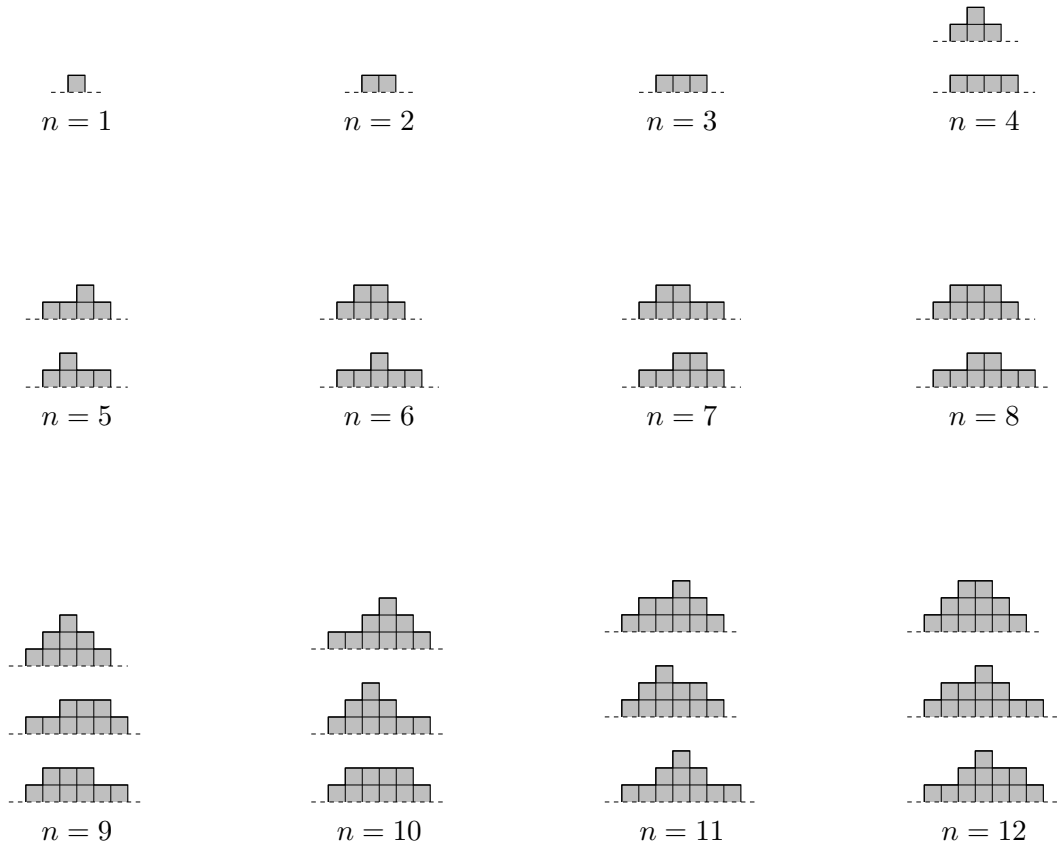
-
- [37] P. KÛRKA : *Topological and symbolic dynamics*, volume 11 de *Undergraduate texts*. Société Mathématique de France, Paris, 2003.
- [38] P. KÛRKA et A. MAASS : Limit sets of cellular automata associated to probability measures. *Journal of Statistical Physics*, 100(5-6):1031–1047, 2000.
- [39] M. LATAPY, R. MANTACI, M. MORVAN et H. D. PHAN : Structure of some sand piles models. *Theoretical Computer Science*, 262:525–556, 2001.
- [40] M. LATAPY et H. D. PHAN : The lattice structure of chip firing games and related models. *Physica D*, 155:69–82, 2001.
- [41] P. B. MILTERSEN : Two notes on the computational complexity of one-dimensional sandpiles. Rapport technique RS-99-3, BRICS, 1999.
- [42] C. MOORE et M. NILSSON : The computational complexity of sandpiles. *Journal of Statistical Physics*, 96(1-2):205–224, 1999.
- [43] H. D. PHAN : *Structures ordonnées et dynamiques de piles de sable*. Thèse de doctorat, Université Paris VII, 1999.
- [44] P. RUELLE et S. SEN : Toppling distributions in one-dimensional Abelian sandpiles. *Journal of Physics A*, 25(22):L1257–L1264, 1992.
- [45] A. R. SMITH III : Real-time language recognition by one-dimensional cellular automata. *Journal of Computer and System Sciences*, 6(3):233–253, 1972.
- [46] J. SPENCER : Balancing vectors in the max norm. *Combinatorica*, 6:55–65, 1986.
- [47] K. SUTNER : Linear cellular automata and de Bruijn automata. *Mathematics and Its Applications*, 460:303–320, 1999.
- [48] S. TAKESUE : Staggered invariants in cellular automata. *Complex Systems*, 9:149–168, 1995.
- [49] D. L. TURCOTTE : Self-organized criticality. *Reports on Progress in Physics*, 62:1377–1429, 1999.
- [50] S. M. ULAM : Random processes and transformations. In *Proceedings of the International Congress of Mathematics*, volume 2, pages 264–275, 1952.
- [51] J. VON NEUMANN : The general and logical theory of automata. In *Cerebral Mechanisms in Behavior: The Hixon Symposium*, pages 1–41, 1951.

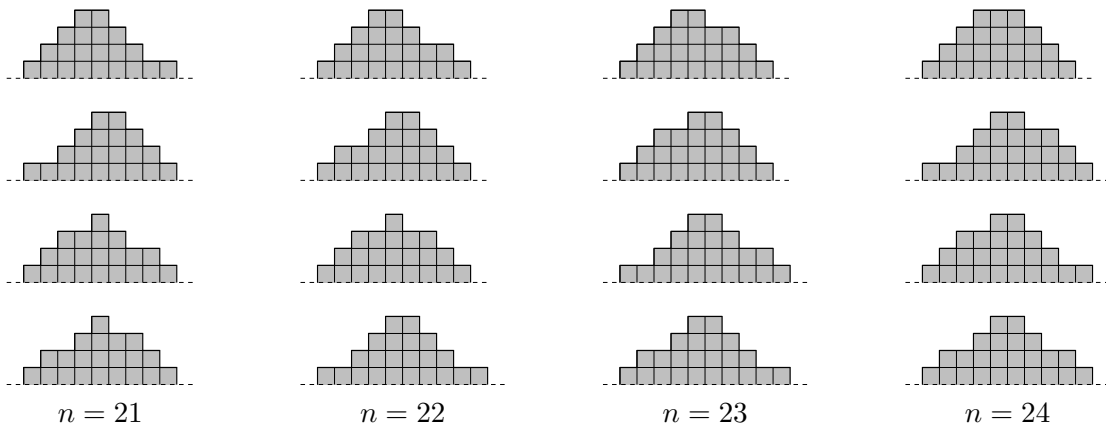
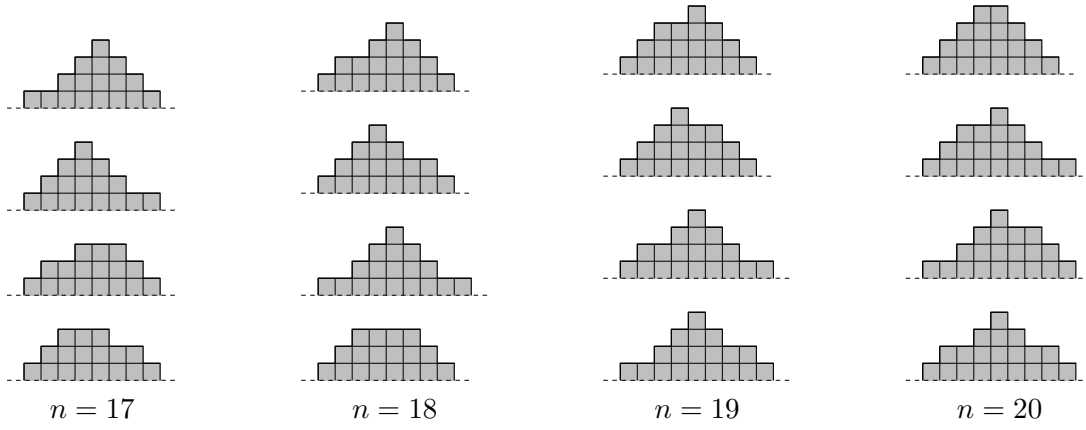
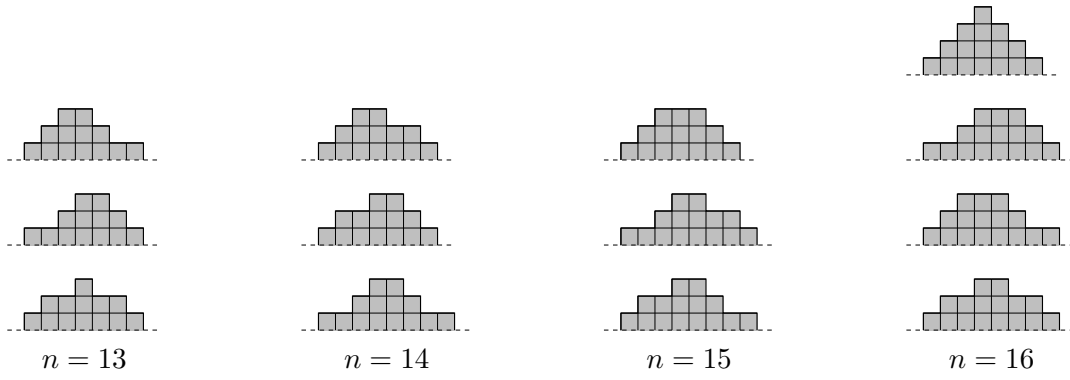
Annexes

A

Points fixes de (n) pour SSPM ($1 \leq n \leq 24$)

Pour la définition de SSPM et de ses points fixes, se référer à la partie I, chapitre 3.





Index

Les numéros de page en **gras** correspondent aux définitions.

- AC, **53**
- AS, **52**
- ascenseur, **75**
- atteignable, **10**
- automate cellulaire, 47, 53, **53**, 54
- automate de sable, 2, 47, 52, **52**, 57

- bijektivité, 58
- boule ouverte, **50**, 51

- cassé, **30**
- centre, **38**
- CFG, 1
- chaînes modulaires, **12**
- chaoticité, 59, 60
- Chip Firing Game, 1, 15
- colonne, **8**, **48**
- compacité, 50
- compacité locale, 51
- complet, 51
- configuration, **8**, **28**, 48, **48**
- conservation des grains, 59, 68, **69**, 73
- conservation des infinis, **56**
- cylindre, **49**, 52

- décalage horizontal, **56**
- décalage vertical, **56**
- décomposition entière, **11**
- diagramme de Ferrers, **8**
- distance, **50**
- droite, **30**

- énergie, **29**
- équicontinuité, 59
- état stable, **10**
- expansivité, 59

- falaise, **8**
- fermé, 51

- FGC, 69, **69**
- finie, **60**, 69

- gauche, **30**
- GC, **69**, 73
- graphe des orbites, 2, 10, **10**, 23, **28**, 29, 30, 33, 35
- graphe des orbites bornées, **17**, 18

- HV-chaînes, **12**

- Ice Pile Model, 1, 7, **9**
- injectif, **60**
- injectivité, 59, 60, 68
- invariante horizontalement, **56**
- invariante verticalement, **56**
- IPM, 1, 7

- longueur du transitoire, 12, **12**, 13, 24, 38, **38**, 40, 42, **74**
- LR-décomposition, **30**
- LR-décomposition cassée, **30**, 33, 35

- machine à deux compteurs, **74**
- mal formée, **79**

- nilpotence, 60

- observateur, **49**
- orbite, **10**
- outil de mesure, **49**

- parfait, 50
- partition d'entiers, 9
- périodique, **60**, 69
- PGC, 69, **69**
- pile de sable, 1, 7, **8**, **28**, 47
- plateau, **8**
- point de référence, **49**
- point fixe, **10**, 11, **28**, 35, 37, 93

portée, 52, **52**
précision, **48**
puits, **48**

quasi-équicontinuité, 59

rayon, 52, **52**
règle globale, **28**, **52**, 57
règle horizontale, **9**
règle locale, **52**
règle verticale, **8**
régularité, 59

Sand Pile Model, 1, 7, **8**
Self-Organized Criticality, 1, 7
sensibilité aux conditions initiales, 59
SIPM, 28, 43
SOC, 1, 7
sommet, **30**
source, **48**
SPM, 1, 7, **8**, 27, 52
SSPM, 3, 27, **28**, 93
surjectif, **60**
surjectivité, 59, 60, 68
Symmetric Ice Pile Model, 28, 43
Symmetric Sand Pile Model, 2, 27, **28**
système dynamique discret, 1, 47

table de transitions, **52**, 56
topologie, 48
totalement déconnecté, 51
transitivité, 59
treillis, 10, 18, 23

ultimement périodique, 59, 74, **74**, 81
ultimement stable, 60, 74, **74**, 82
ultramétrique, 50

valide, **79**
voisinage, **53**, **71**

Résumé

Dans cette thèse nous étudions différents systèmes dynamiques discrets permettant de simuler la formation des piles de sable. Le comportement des modèles de base SPM ou $IPM(k)$ est bien connu dans des conditions initiales spécifiques. Nous étendons ces résultats à des conditions initiales plus générales, et nous introduisons le modèle SSPM qui ajoute de la symétrie à ces modèles et améliore leur réalisme.

Dans un second temps, nous étudions un autre système dynamique, les automates de sable. Ils sont définis de manière analogue aux automates cellulaires, avec la contrainte supplémentaire qu'une configuration n'admet pas de « trous ». Ces automates peuvent simuler tous les modèles de piles de sable définis localement, et à l'aide d'un cadre mathématique solide, ils permettent d'obtenir des résultats plus généraux.

Nous nous intéressons à la dynamique des automates de sable, plus précisément aux propriétés de réversibilité d'un automate, et nous étudions la décidabilité de propriétés caractérisant les piles de sable classiques : conservation des grains et périodicité ultime.

Mots-clés : Systèmes dynamiques discrets, piles de sable, Sand Pile Model, Ice Pile Model, automates de sable, automates cellulaires, réversibilité, décidabilité.

Abstract

In this thesis we study several discrete dynamical systems which can simulate the formation of sandpiles. The behavior of the basic models SPM and $IPM(k)$ is well-known under specific initial conditions. We extend these results to arbitrary initial conditions. Moreover, we introduce the model SSPM which adds symmetry to these models and improves their realism.

In the second part, we study another dynamical system, namely sand automata. They are defined similarly to cellular automata, with the constraint that configurations can not contain any “hole”. These automata can simulate any locally-defined sandpile model, and their solid topological framework allows more general results.

We are interested in sand automata dynamics, more precisely in the reversibility properties of sand automata. We conclude by studying the decidability of properties which characterize classical sandpile systems: grain conservation and ultimate periodicity.

Keywords: Discrete dynamical systems, sandpiles, Sand Pile Model, Ice Pile Model, sand automata, cellular automata, reversibility, decidability.